

Bridging Pretraining and Post-Training: Toward Reasoning-Centric Large Language Models

Syeda Nahida Akter

CMU-LTI-26-006

May 2026

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Eric Nyberg (chair), Carnegie Mellon University
Chenyan Xiong, Carnegie Mellon University
Sean Welleck, Carnegie Mellon University
Yejin Choi, Stanford University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Language and Information Technology.

To my loving parents, my wonderful sister, and my super-cool husband—thank you for your endless love and support.

Acknowledgments

This thesis reflects the guidance, encouragement, and kindness of many people who have supported me throughout my doctoral journey. I am deeply grateful to each of them.

First and foremost, I would like to thank my advisor, Eric Nyberg, who believed in me from the very beginning. Eric saw potential in me even when I was still learning to see it in myself. I am profoundly grateful for his guidance, thoughtful feedback, and unwavering support over the years. His patience, especially during moments when I struggled or made repeated mistakes, taught me important lessons about humility and perseverance. Even during the most stressful periods of graduate school, a conversation with Eric would somehow make the pressure feel lighter. He never expected me to be a paper-machine; instead, he constantly reminded me that meaningful research values quality over quantity. That principle has shaped how I approach every project. As a married PhD student whose partner lived in another state for much of this journey, balancing life and research was not always easy. Eric's flexibility and understanding made it possible for me to navigate those challenges with peace of mind. Professor Eric, I will always look up to you not only as an exceptional mentor but also as a remarkable human being. The sparkle in your eyes when discussing a new idea is a gift that has left me forever inspired.

I am also sincerely thankful to my committee members: Yejin Choi, Chenyan Xiong, and Sean Welleck. Each of them played an important role in shaping my growth as a researcher. Conversations with Yejin during the early stages of my research taught me how to think about ideas that are both impactful and meaningful to the broader community. Her approach to research vision and her ability to frame ideas through clear storytelling have influenced the way I write papers and present my work today. I am grateful to Chenyan Xiong for his careful guidance and thoughtful feedback during my early projects, which helped refine my research questions and sharpen my thinking. I would also like to thank Sean Welleck for his insightful feedback on this thesis, which greatly improved the clarity and depth of the work. During the early years of my PhD, I was fortunate to receive guidance from Louis-Philippe Morency and Yonatan Bisk as well. Their encouragement and mentorship helped shape my approach to critical thinking and experimental design.

I am deeply grateful to my colleagues at NVIDIA, who have supported and encouraged me throughout this journey. In particular, I would like to thank my mentor Shrimai Prabhunoye. Our many brainstorming sessions and discussions on research ideas, problem formulation, and experimental strategies have been invaluable to my development as a researcher. I never imagined that an internship would lead me to meet someone so thoughtful, supportive, and enthusiastic about research. Shrimai's encouragement and perspective

continue to inspire me. I am also thankful to my manager Mostofa Patwary, whose trust and belief in me have meant a great deal. My sincere thanks to Ali Hatamizadeh for encouraging curiosity and teaching me to think beyond conventional boundaries. I also appreciate the support and feedback from Mohammad Shoeybi and Bryan Catanzaro, whose insights helped strengthen several of the projects included in this thesis.

On a more personal note, I would like to thank the friends who made this journey less lonely and far more meaningful. As someone who tends to be quiet and reserved, making new connections has never come easily for me. Yet I have been fortunate to meet people whose kindness and companionship brought warmth into my life. Yingshan Chang, Aman Madaan, Alex Wilf, and Leena Mathur have been wonderful friends whose presence and support meant more than words can fully express. Yingshan, your words have lifted my spirits on countless occasions, and I am grateful for your friendship. I am also thankful to reconnect with my undergraduate friends here in the United States—Bristy, Sazan, and Oaishi. Because of you, I stepped outside my apartment and created some of my most cherished memories in Pittsburgh. My heartfelt thanks also go to all my friends from Eccentric 11 and the Papiya group for patiently listening to my endless complaints and worries.

Finally, and most importantly, I owe everything to my family. Their love and support have been the foundation that carried me through this journey. Even while living far away from home, I always found comfort in knowing that I could call my parents or my sister Zidni whenever I needed encouragement. My father, Nizam, celebrated even my smallest achievements with pride and excitement, which gave me the strength to keep moving forward. My mother, Jabun, has made countless sacrifices for me, and her resilience continues to inspire me every day. My sister Zidni has a special way of brightening even the most difficult days with her silly stories and laughter. Twisha and Amy, you are my shining stars, and life would feel incomplete without you both.

And finally, to my partner, Saiful—thank you for standing beside me through every high and low of this journey. Your patience, encouragement, and quiet strength have been my anchor. You have been the calm in my storm, and I will always be grateful for your love and support. I love you for eternity.

Abstract

The rapid progress of large language models has transformed how machines process and generate information. Despite their scale, many models still struggle with structured reasoning, often relying on superficial correlations rather than systematic inference. This thesis studies how reasoning abilities can emerge in generative models and identifies three factors that shape the emergence of reasoning in large language models: the structure of the data used to teach reasoning, the stage of training at which reasoning supervision is introduced, and the learning objectives that encourage models to form intermediate reasoning processes.

I first investigate the anatomy of high-quality data for reasoning. I analyze what distinguishes data that supports structured inference from data that merely increases training volume. I introduce a novel dataset for complex visual reasoning that includes unanswerable questions requiring verification of image-question consistency. To study how multimodal capabilities can assist textual reasoning, I introduce a modular approach that allows a vision-language model to generate visual representations of text problems and reason over them without additional supervision. I further present a scalable framework that transforms web documents into structured dialogue-style reasoning data, and demonstrate that such conversational reasoning traces significantly improve pretraining efficiency. I then extend self-learning beyond mathematics to multiple reasoning domains by curating structured cross-disciplinary datasets. The second part of the thesis examines how reasoning data interacts with different stages of the training pipeline. Through the first systematic investigation of how SFT-style chain-of-thought data interacts across training phases, I uncover an asymmetric allocation principle: diversity and scale govern pretraining, while quality governs SFT. Front-loading reasoning data into pretraining creates foundations that post-training alone cannot recover. The final part proposes a redesign of the learning pipeline to enable reasoning as an emergent pretraining capability rather than a post-hoc skill. I introduce a continued pretraining strategy that automatically prioritizes masking tokens most distinctive to the target domain. Further, I present a novel pretraining approach which embeds reinforcement learning directly into the pretraining objective by rewarding chain-of-thought generation based on its predictive utility.

To summarize, this thesis provides a unified end-to-end perspective on cultivating reasoning throughout the entire model development lifecycle.

Contents

1	Introduction	1
1.1	Background	2
1.1.1	The Importance and Challenges of Pretraining Data Curation	2
1.1.2	Synthetic Data for Augmenting Pretraining Quality	3
1.1.3	Emergence of Reasoning Ability in Pretraining	4
1.2	Thesis Statement	5
1.3	Overview	6
1.4	Chapter Overview	8
I	Anatomy of High Quality Data for Reasoning	11
2	High Quality Data for Generative Models	12
2.1	Defining Attributes of High-Quality Data for Reasoning	13
2.2	Problem Statements	14
2.2.1	Visual Reasoning	14
2.2.2	Unimodal Reasoning with Multimodal Models	15
2.2.3	Structured Data Synthesis for Efficient Pretraining	15
2.2.4	Generalize across Non-math Reasoning Domains	16
3	Data Synthesis for Improved Visual Reasoning	18
3.1	Task Overview	18
3.2	Data Generation	19
3.2.1	Scene Graph Processing	20
3.2.2	Dataset Analysis and Comparison	21
3.3	Visual Reasoning with Structured Data	24
3.3.1	Pseudocode Generation	24

3.3.2	Pseudocode-Guided Reasoning	25
3.4	Experiments and Analysis	25
3.4.1	Baseline Experiments	25
3.4.2	Analysis	27
3.5	Related Works	28
3.6	Summary	29
4	Unimodal Reasoning with Multimodal Models	30
4.1	Task Overview	30
4.2	Methodology	31
4.2.1	Generate Image from Question	32
4.3	Experiments	32
4.3.1	Experimental Setup	32
4.3.2	Results	33
4.4	Ablations	35
4.4.1	Math Reasoning	35
4.4.2	Symbolic Reasoning	37
4.5	Related Works	39
4.6	Summary	40
5	Structured Data Synthesis for Efficient Pretraining	41
5.1	Task Overview	41
5.2	Methodology	43
5.2.1	Compose Conversational Prompts	44
5.2.2	Generate Conversation	45
5.3	Experimental Setup	45
5.3.1	Training Details	46
5.3.2	Evaluation Metrics	46
5.4	Experiments and Results	47
5.5	Ablations	49
5.6	Discussion: Why Role-Based Conversations Work?	53
5.7	Related Works	54
5.8	Summary	56

6	Scaling Self-Learning beyond Math Reasoning	57
6.1	Task Overview	57
6.2	Methodology	59
6.3	Experiments and Results	62
6.3.1	Experimental Setup	62
6.3.2	Results	63
6.4	Ablations	66
6.5	Related Work	69
6.6	Summary	70
II	Effect of reasoning data across LLM development phases	71
7	Synergy between Pretraining and Post-Training Data	72
7.1	Task Overview	73
7.2	Methodology	74
7.2.1	Model Architecture and Baseline	75
7.2.2	Data Pipeline	75
7.2.3	Synergy between Pretraining and SFT	75
7.3	Experiments and Results	77
7.3.1	Training	77
7.3.2	Evaluation Metrics	77
7.3.3	Results	78
7.4	Ablations	80
7.5	Related Work	83
7.6	Discussion: Mechanistic signatures of front-loading reasoning data	85
7.7	Summary	90
III	Redesign the Learning Pipeline to Enable Reasoning Emergence	91
8	Bridging Gap between Pretraining and Post-training	92
8.1	Problem Statements	93
8.1.1	Difference Masking in Pretraining	93
8.1.2	Reinforcement as Pretraining Objective	94

9	DIFFERENCE-MASKING: Choosing What to Mask in Continued Pretraining	95
9.1	Task Overview	95
9.2	Methodology	97
9.2.1	Problem Setting	97
9.2.2	Motivation and Notation	97
9.2.3	Our Approach: DIFFERENCE-MASKING	97
9.2.4	Finding Difference Anchors: TF-ICF	98
9.2.5	Masking Based on Differences	98
9.3	Experimental Settings	99
9.3.1	Datasets	99
9.3.2	Baseline Methods	100
9.3.3	Experimental Methodology	101
9.4	Results and Discussion	101
9.4.1	Comparison with Baseline Approaches	101
9.4.2	What is masked?	103
9.4.3	Sensitivity Analysis	104
9.5	Related Work	105
9.6	Summary	106
10	Reinforcement as Pretraining Objective	107
10.1	Task Overview	107
10.2	Methodology	108
10.2.1	Reasoning as an action	109
10.2.2	Information-gain reward	110
10.2.3	Expected improvement identity	110
10.2.4	RLP objective and optimization	111
10.2.5	Reward properties and guarantees	111
10.3	Experimental Setup	112
10.3.1	Evaluation Metrics	113
10.4	Results	114
10.5	Ablations	116
10.5.1	How RLP Impacts the Base Model Representation Across LLM Families	118
10.6	Related Work	123

10.7 Summary	123
IV Discussion and Future Directions	124
11 Conclusion	125
11.1 Summary of Contributions	125
11.2 Future Directions	127
11.2.1 Data Anatomy in Model Development	127
11.2.2 Effective Integration of High Quality Data	130
11.2.3 Rethinking Training Pipeline	131
11.2.4 Reasoning Bases for Long-Horizon and Agentic Tasks	133
11.3 Broader Impact	134
12 Appendix for Anatomy of Data	174
12.1 Data Balancing	174
12.2 Overview and Analysis of the VISREAS	175
12.3 Baseline Configuration	177
12.4 Effect of pseudocode finetuning	179
12.5 Examples from VISREAS and GQA	180
12.6 Mechanical Turk Details	180
12.7 SELF-IMAGINE Prompts	181
12.8 Prompts and Datasets for MIND	183
12.8.1 Prompts for Conversation	183
12.8.2 Evaluation Metric Details	186
12.9 Additional Experiments and Results	190
12.9.1 Results of Pretraining LLM from Scratch	190
12.9.2 Breakdown of Individual Tasks Results of Continued Pretrained LLM	193
12.10 Additional Ablations	194
12.10.1 Context Length vs Conversation Quality	194
12.10.2 Conversation Length vs Model Performance	195
12.10.3 Conversation Quality Assessment	195
12.10.4 Compare with DEEPSEEK MATH	197
12.10.5 Conversations on Code Tasks	198

12.11 Dialogue Examples	205
12.12 Data Proportion across Blends	217
12.13 Token Efficiency Analysis	218
12.14 Sub-category Accuracy Analysis	219
12.15 Relative Importance of Structure and Context: Evidence from Ablations	220
13 Appendix for Effect in Data Integration	225
13.1 Experiments and Results	225
13.2 Additional Ablations	228
14 Appendix for Redesigning Pipeline	233
14.1 Detailed Experimental Settings	233
14.2 Masking Video Tokens	234
14.3 Masking Language Tokens	234
14.3.1 Proofs	235
14.3.2 Proof of Proposition 1 (Expected improvement identity)	235
14.3.3 Proof of Proposition 2 (Lower bound via marginalization over thoughts)	236
14.3.4 Tokenwise-to-sequence connection under teacher forcing (position-wise credit)	236
14.4 Why Relative Advantages Do Not Reward Bad Thoughts	238
14.4.1 Proof of Monotonic Improvement	238
14.4.2 Numerical Illustration of Relative Advantage Updates	239
14.5 Experimental Setup	241
14.6 Extended ablation details	242
14.7 Additional Ablations	243
14.8 Data Blend Extended Results	250

List of Figures

3.1	Overview of VISREAS dataset construction process. Using scene graphs, we cluster objects (orange), relations (green), and attributes of the related objects (blue) based on the attribute of the corresponding objects (orange). Then the question engine takes each template as input and traverses all possible clusters to generate the query as well as the reasoning steps. Each function in the reasoning steps can return NONE if any object, attribute, or relation is absent in the image.	19
3.2	Semantic string parser. For every line of semantic string, we use regex and string manipulation to extract operator and its arguments. We represent scene-graph in adjacency list format and run the parsed operator to get formatted pseudocode and its expected output.	22
3.3	Overview of VISREAS statistics. (Top left) The dataset covers 14 attributes in a balanced ratio. (Top right) It consists of five reasoning types of queries in a balanced distribution. (Bottom left) Comparison of multi-hop relation traversal for different VQA datasets. Majority questions of VISREAS require multi-hop traversal compared to others. (Bottom right) Comparison of number of objects mentioned in the question for different datasets where VISREAS questions contain larger amount of objects.	23
3.4	VISREAS contains two types of relation traversals. Star relation states a single object that shares multiple relations with other objects (Left). Chain relation states multiple objects that share a single relation with each other (Right).	23
3.5	Overview of LOGIC2VISION. In Pseudocode Generation phase, we generate pseudocode which outlines the reasoning steps. During Pseudocode-Guided Reasoning , the pseudocodes along with the question and image are provided to the model. The model executes all intermediate pseudocodes to arrive at the final answer.	24
4.1	Generating an image from a question via a single VLM through HTML.	31

4.2	[Left] Reasoning using VLM without : Given a question (0), the VLM generates an answer (1). [Right] Reasoning using VLM with : Given a question (0), the VLM generates a structured representation of the question using HTML (1). The HTML is rendered as an image (2) which is then passed along with the question to the VLM again (3). The VLM finally generates the answer by combining both vision and language modalities (4).	31
4.3	SELF-IMAGINE main results: SELF-IMAGINE improves accuracy over a diverse range of mathematical and symbolic reasoning tasks.	33
4.4	Example from math world problem tasks.	35
4.5	Examples from some BIG-Bench Hard sub-tasks.	38
5.1	Continuous pretraining with all styles of conversations (MIND-OWM-4B) derived from a small subset (OWM-4B) and a $3.6\times$ large raw corpus (OWM-14B) reveals that model trained with conversations outperforms the one trained with larger corpus in GSM8K, MMLU and general reasoning—showing the significance of high-quality structured data over quantity.	42
5.2	Math Informed syNthetic Dialogue. We (a) manually design prompts of various conversational styles, (b) provide the prompt along with raw context as input to LLM to obtain diverse synthetic conversations, (c) apply heuristic filtering to refine the generated data and (d) observe the downstream task performance after continuously pretraining an 7B LLM.	43
5.3	Similarity between Raw Text & Synthetic Dialogues. The TWO PROFESSORS style exhibits greater similarity to raw text, while LAYMAN KNOWALL shows the lowest similarity due to its richer context with details and explanations.	49
6.1	Employing self-learning with multi-domain data, NEMOTRON-CROSTHINK outperforms baseline models, including domain-specific training (Only Math) and Open-Reasoner-Zero (ORZ-7B), achieving consistent gains across all reasoning tasks.	58
6.2	NEMOTRON-CROSTHINK. We (a) curate QA pairs from synthetic (Common Crawl) and open-source datasets, categorized into general-purpose reasoning (\mathcal{D}_{gpr}) and mathematical reasoning (\mathcal{D}_{mr}); (b) apply structured templates to convert data into multiple-choice (MCQ) and open-ended formats, promoting diverse reasoning trajectories; (c) filter out unverifiable or ill-formatted responses; (d) train an RL policy using Group Relative Policy Optimization (GRPO). The final reward is used to update the policy, iteratively improving the model’s reasoning capabilities across diverse domains.	59
6.3	Token efficiency comparison of models trained on $\mathcal{B}_{gpr\uparrow}$ (multi-domain blend) and two single domain blends (\mathcal{B}_{only_math} and ORZ).	66

7.1	We systematically inject reasoning-style data (\mathcal{D}_{res}) at different phases of training—pretraining versus SFT—while varying its <i>diversity, quantity, and quality</i> . Our results show an asymmetric principle: diversity and scale matter most during pretraining, whereas quality dominates in SFT. This allocation strategy compounds through reinforcement learning (RL), yielding sustained gains across complex reasoning benchmarks.	74
7.2	Layerwise comparison between FLR and Base. Top: CKA similarity per layer. Middle: mean hidden-state L2 difference. Bottom: hidden-state variance ratio.	86
7.3	Attention-head level CKA between FLR and Base across layers. Lower similarity indicates stronger head-level divergence.	87
7.4	Summary statistics of attention-head divergence between FLR and Base. Top: mean CKA across heads per layer. Bottom: spread across heads, where higher spread indicates that divergence is concentrated in a subset of heads.	87
7.5	Hidden-state norm and variance per layer for FLR and Base.	88
7.6	Output-space comparison between FLR and Base across generation positions. Top: KL divergence. Middle: Jensen–Shannon divergence. Bottom: logit correlation.	89
7.7	Token-level prediction entropy by generation position for FLR and Base, and their difference. Negative values indicate that FLR is more confident.	90
9.1	DIFFERENCE-MASKING automatically selects <i>what to mask</i> based on what makes the task domain <i>different</i> from the pretraining domain, enhancing model learning on the end task.	96
9.2	DIFFERENCE-MASKING: an approach to choosing what to mask during continued pretraining that prioritizes masking concepts that make the target domain different from the pretraining domain. DIFFERENCE-MASKING does this by first selecting <i>anchor topics</i> relating to the downstream task, and then by masking words or bounding boxes based on their similarity to those anchor topics.	96
9.3	The most frequently masked words chosen by the DIFFERENCE-MASKING algorithm across the ChemProt and ACL-ARC tasks. We find that for the ChemProt dataset, the masks we find automatically through unlabelled data partially recover the end task labels.	102
9.4	Performance on both tasks is best at the hyperparameter $K = 20$ anchors.	105
10.1	Visualization of the RLP framework. A chain-of-thought is sampled <i>before</i> next-token prediction. Rewards are computed by contrasting the predictor conditioned on the CoT with a <i>No-think</i> EMA baseline, yielding a verifier-free, dense signal. We list the advantages of RLP over the traditional pretraining objective (top right) and show the impact after end-to-end training (top left).	109
10.2	Ablations on Qwen3-1.7B. Curves report Math/Science/Overall averages. Dashed lines mark the base model.	117

10.3	Representation change after RLP, by layer. Each column is one model family: CKA between base and RLP hidden states (top), mean per-element L2 $\ \mathbf{h}_{\text{RLP}} - \mathbf{h}_{\text{base}}\ _2$ (middle), and hidden-state variance ratio $\text{Var}(\mathbf{h}_{\text{RLP}})/\text{Var}(\mathbf{h}_{\text{base}})$ (bottom). Same prompts and pooling over positions for all panels.	119
10.4	CKA per head (base vs. RLP). Each panel is a layer \times head heatmap (yellow \approx identical subspaces, purple/blue \approx strong divergence). Same prompts and pooling as Fig. 10.3.	122
11.1	Per-question coverage of reasoning phrases (normalized) in AIME-25 solutions from four LLMs. Each axis is one phrase (verify, wait, reconsider, check again, misunderstanding, re-examine). Larger area indicates broader and higher use of these phrases. The ranking by coverage aligns with AIME-25 accuracy: Nemotron 5-8B-base > Qwen 3-8B-base > Qwen 2.5-7B-base > LLaMA 3.1-8B-base.	129
12.1	Overview of types of questions along with some templates and examples from the VISREAS corpus.	175
12.2	Distribution of VISREAS questions semantic length (number of computation steps to arrive at the answer) as well as the readability scores for each semantic step type. We can see that most questions require at most two reasoning steps, where each step may involve tracking a relation between objects, an attribute identification, or a logical operation. At the same time, questions with larger semantic steps are difficult to read.	176
12.3	(a) Question distribution across the number of attributes in a query. The question complexity increases with the number of attributes or relations. (b) Average question length per reasoning type in VISREAS corpus. Query questions are lengthier than other reasoning categories as these questions contain multiple objects of similar attributes with their relations.	177
12.4	Pseudocode format. Our method re-structures the format of GQA semantic string to pseudocode to better leverage Code-LLMs without adding any auxiliary information.	177
12.5	Overview of pseudocodes for two different traversal types in the VISREAS corpus.	178
12.6	Human accuracy on different attributed questions	178
12.7	Example questions from the VISREAS and the GQA corpuses.	181
12.14	With increasing context length the generated conversation length decreases!	194
12.15	LLM tends to rate its generation higher most of the times.	196

12.8 Amazon Mechanical Turk interfaces used for Qualification Test to choose the right workers for human accuracy assessment on VISREAS task. We study the workers by deploying two tasks. In the first task, we ask the workers to read the instructions carefully (Top left) and answer some multiple-choice questions (Top right). After passing this task, ten questions with images will be presented and the final task would be to choose the right answer from the answer dropdown list (Bottom right). We choose the workers for the final evaluation who have correctly predicted more than seven answers out of ten questions.	200
12.9 Amazon Mechanical Turk interfaces for human accuracy assessment on VISREAS task using the qualified workers. (a) For each HIT, we provide an image and a question that needs to be answered from a dropdown list of 2013 entries. In addition, we ask for rating the complexity and structural soundness of the query and further look for details if any Turker finds the question problematic. (b) To investigate what type of problem the question possesses, we ask for further details from the workers and even encourage them to rewrite the query to remove the problem they faced while answering the query.	201
12.10 Accuracy by question length across three mathematical reasoning tasks. In the cases of ASDIV and SVAMP, accuracy is notably higher when utilizing images for longer and more intricate questions compared to scenarios without images. However, in the context of more complex questions, such as those found in GSM8K, the limitations of the VLM become apparent. In this scenario, the inability to generate effective HTML results in erroneous image generation, consequently leading to decreased accuracy, particularly with longer questions.	202
12.11 GSM8K accuracy by chain-of-thought length. Similar to the findings in Figure 12.10, image representations for complex questions are not efficient and structured. Therefore, the inclusion of images does not enhance the representation of questions that demand longer chains of thought.	203
12.12 Accuracy by Question Length for a subset of BIG-Bench-Hard benchmark. Incorporating images helps when the corresponding question is simpler and shorter and when the questions are more complex.	203
12.13 Number of Instances from each subtask impacted by Image. Here ‘ <i>Image Hurts</i> ’ represents instances that achieved correct answers without image and got incorrect with image. Similarly ‘ <i>Image Improves</i> ’ shows data points getting the correct answers with image and getting incorrect without image.	204
12.16 Average token lengths of correct and incorrect responses across general-purpose and math reasoning tasks for models trained on $\mathcal{B}_{gpr\uparrow}$, \mathcal{B}_{only_math} , and ORZ.	218

12.17	Sub-category Accuracy Comparison across MMLU-PRO Domains. The $\mathcal{B}_{gpr\uparrow}$ blend consistently outperforms \mathcal{B}_{only_math} in a wide range of non-math reasoning categories such as business, law, psychology, and economics. Surprisingly, it also slightly surpasses the math-specialized blend in the MMLU-PRO math category, highlighting the generalizability and versatility of multi-domain training.	219
12.18	Sub-category Accuracy Comparison across AGIEVAL. While \mathcal{B}_{only_math} performs marginally better in the math, $\mathcal{B}_{gpr\uparrow}$ achieves stronger results in non-math domains.	219
12.19	Sub-category Accuracy Comparison across SUPERGPQA. The $\mathcal{B}_{gpr\uparrow}$ blend consistently outperforms \mathcal{B}_{only_math} in a wide range of non-math reasoning categories except the science category which consists of fields like mathematics, physics, astronomy, chemistry etc.—highlighting the generalizability and versatility of multi-domain training.	220
12.20	Guided Multi-Step Reasoning Expansion.	221
13.1	The model that saw the same high-quality data in both pretraining and SFT (\mathcal{M}_{SHQ}) handily beats the baseline (\mathcal{M}_{base}) that only saw the data once. . . .	230

List of Tables

3.1	Question-template distribution over attributes	21
3.3	Accuracy breakdown of baseline models and humans on VISREAS across different reasoning types. Problematic questions contain relations, attributes, or objects that are missing or inconsistent with the image, whereas Non-Problematic questions are consistent with the image. Except for the Code-GEN models, we report fine-tuned results on VISREAS.	26
3.2	Performance comparison among baseline models on GQA and VISREAS. (*) GQA trainset images were used during training.	26
3.4	Breakdown of accuracies on VISREAS for 's VICUNA model size. We observe that VICUNA's model size improves performance in most question-types except the problematic ones.	27
4.1	Comparison of accuracy between 'Question Only' and 'Question + Image' across diverse reasoning tasks where the image has been generated using SELF-IMAGINE.	34
4.2	Example of Image improving reasoning in GSM8K task for LLAVA-1.5. . . .	36
5.2	Results of 7B LLM trained on Complete OWM-14B and MIND-OWM-14B: Continuous training of LLM with synthetic conversation outperforms models trained with original pretraining blend and raw OpenWebMath across all tasks.	48
5.3	TWO PROFESSORS prompt style vs Raw data. Continuous pretraining with TWO PROFESSORS conversations does not provide gain over raw data compared to other conversational styles.	49
5.4	MATHPILE vs Synthetic Conversation from MATHPILE (MIND-MATHPILE). Conversation generated from high-quality raw data further improves the performance of math tasks.	50
5.5	Comparison of 7B LLM trained with raw and combination of synthetic data. Synthetic conversation outperforms raw data in all combinations. Specifically, combinations of all conversations generated from OWM-4B surpasses the performance of OWM-14B (3.6× larger corpus) across all tasks, underscoring the superior quality and diversity of the conversations.	51

5.6	Results of 7B LLM trained on MIND-OWM-4B using \mathcal{M} of different sizes: Regardless of the sizes of \mathcal{M} , model trained on MIND-OWM-4B outperforms the one trained with raw data.	52
6.1	Training data distribution by source and type. OE=Open-Ended; GPR =General-Purpose Reasoning; MR =Math Reasoning	60
6.2	Overview of Data Blending Strategies. Blends are categorized by data source, question type, and usefulness—each constructed to assess the im- pact of domain diversity, format variation, and task relevance on RL-based reasoning.	61
6.3	Results of Self-Learning on Individual Datasets. Each row shows the downstream evaluation results after self-learning on a single data source. Results highlight the varying strengths of individual datasets across general- purpose and mathematical benchmarks.	64
6.4	Results of NEMOTRON-CROSTHINK-7B across Blends. Multi-domain blend $\mathcal{B}_{gpr \uparrow}$ achieves the highest overall average accuracy, outperforming domain-specific and naturally sampled blends—underscoring the benefit of self-learning with diverse reasoning data. (*) Due to the space shortage, we use *CROSTHINK to refer NEMOTRON-CROSTHINK.	65
6.5	Impact of Question Format. Converting all questions to open-ended format improves accuracy across benchmarks, reducing reliance on option guessing and encouraging deeper reasoning.	67
6.6	Impact of Answer Format. Using short-form answers improves accuracy by reducing output ambiguity and avoiding penalization from rigid reward functions in rule-based training.	67
6.7	Difficulty-Based Filtering. Filtering $\mathcal{B}_{gpr \uparrow}$ to retain only hard examples ($\mathcal{B}_{f(gpr \uparrow)}$) yields consistent gains across all tasks, highlighting the effective- ness of selective training on challenging data.	68
7.1	Average Accuracies of base models trained without or with varying \mathcal{D}_{res}. Pretraining with diverse reasoning data yields immediate gains, with scale and diversity driving math and code improvements, more than quality. \mathcal{M}_{res} represents the average of \mathcal{M}_{SHQ} , \mathcal{M}_{LDQ} , and \mathcal{M}_{LMQ}	79
7.2	Average Accuracies of SFT models pretrained with varying \mathcal{D}_{res}. SFT am- plifies the pretraining advantage—models with reasoning-rich pretraining significantly outperform baseline.	79
7.3	Average accuracies of RL models pretrained and fine-tuned with varying \mathcal{D}_{res}. Introducing reasoning data early provides significant reasoning boost after post-training.	80

7.4	Impact of diverse pretraining \mathcal{D}_{res} on SFT phase. Doubling SFT for the baseline fails to “catch up” to reasoning-pretrained models, while the latent advantage of the mixed-quality pretraining (\mathcal{M}_{LMQ}) emerges, making it the top performer.	81
7.5	Impact of diverse SFT \mathcal{D}_{res} on SFT phase. Fine-tuning on the small, high-quality corpus (\mathcal{D}_{SHQ}) is highly effective, while using large, diverse corpora (\mathcal{D}_{LDQ}) degrades reasoning.	81
7.6	Effect of varying reasoning data ratio during pretraining. Increasing the reasoning proportion improves reasoning-focused benchmarks while preserving general-domain performance.	82
7.7	Effect of pretraining reasoning ratio on downstream SFT performance. Higher reasoning proportion improves reasoning benchmarks but slightly reduces instruction-following metrics.	82
7.8	Impact of scaling reasoning data in SFT phase. Naively doubling mixed-quality data is detrimental to math reasoning, whereas targeted scaling of high-quality data yields consistent gains.	83
9.1	We find that DIFFERENCE-MASKING outperforms strong baselines in both the language and multimodal experimental settings. We note that our entirely self-supervised method also outperforms Selective Masking, which uses labelled data to inform its masking strategy. Values are average results over five trials, subscripts are standard deviations.	99
9.2	For each method, we analyze what percent of tokens are chosen to be masked from within bounding boxes over people as opposed to objects.	104
9.3	Ablating DIFFERENCE-MASKING’s anchor-scoring function based on nearest-neighbor and replacing it with one based on similarity with the anchor embeddings’ centroids leads to performance degradation. This provides evidence for our hypothesis that the nearest-neighbor scoring function helps make DIFFERENCE-MASKING robust to anchor selections.	104
10.1	Quantitative benchmarks for Qwen3-1.7B-Base, showing the impact of RLP. Shaded columns indicate RLP variants; “Post” indicates SFT + RLVR post-training.	113
10.2	Quantitative benchmarks for NEMOTRON-NANO-12B-v2, showing the impact of RLP. Shaded columns indicate RLP variants; “Post” indicates SFT + RLVR post-training.	114
10.3	Token- and flop-matched comparisons of RLP and RPT using a QWEN3-1.7B-BASE model.	115
10.4	RLP across diverse corpora. RLP trained on six SFT-style and general-purpose datasets yields consistent gains, indicating transferable reasoning from mixed/open-ended data.	116

12.1	Hyperparameters for LOGIC2VISION model	179
12.2	LoRA configurations	179
12.3	Hyperparameters of all CLS baselines	179
12.4	Effect of pseudocode finetuning on LLaVA-1.5	180
12.5	Prompts used for both reasoning and mathematics tasks. For all reasoning tasks, we also add <i>Please think step-by-step, and finally answer by selecting an option using the format "The answer is ⟨option⟩"</i> after adding the question to the above mentioned prompts.	184
12.6	Example of Image improving reasoning in GSM8K task for GEMINI PRO.	185
12.7	Example of Image hurting reasoning in ASDIV task for LLAVA-1.5.	186
12.8	Example of Image hurting reasoning in TRACKING SHUFFLED OBJECTS of three objects task for GEMINI PRO.	187
12.9	Example of Image hurting reasoning in NAVIGATE task for GEMINI PRO.	188
12.10	Example of Image improving reasoning in GEOMETRIC SHAPES task for GEMINI PRO.	189
12.11	Example of Image hurting reasoning in GSM8K task for LLAVA-1.5.	190
12.12	Example of Image improving reasoning in NAVIGATE task for LLAVA-1.5.	191
12.13	Example of Image hurting reasoning in DATE UNDERSTANDING task for LLAVA-1.5.	192
12.14	Evaluation of 8B LLM on General Reasoning tasks: Conversations provide improvement over raw data in general purpose reasoning tasks including commonsense, factual and social reasoning tasks.	192
12.15	Evaluation of 8B LLM on Math and Specialized Knowledge tasks: Conversations are always beneficial for mathematical reasoning where specialized knowledge tasks further benefit from having both raw and synthetic data in the corpus.	192
12.16	Results of 7B LLM on General Reasoning Tasks: We evaluate both the baseline and model trained with synthetic data across diverse tasks that focus on general reasoning, language understanding and commonsense.	193
12.17	Results of 7B LLM on Specialized Knowledge Tasks: In this setup, we assess the domain specific knowledge of LLM specifically on mathematics, science and general knowledge. We emphasize on the GSM8K, MATH and MMLU-STEM task, as these tasks predominantly checks the mathematical reasoning ability of the LLM.	193
12.18	Evaluations on General Reasoning Tasks with complete OWM-14B: Conversational data is beneficial for general purpose reasoning tasks.	194
12.19	Evaluations on Math and Specialized Knowledge Tasks with complete OWM-14B: Conversations improve mathematical reasoning over raw data.	194

12.20	Conversation Length vs Downstream Task Accuracy: Conversation length is not correlated with downstream task accuracy.	195
12.21	DEEPSEEK MATH vs All Synthetic Conversations. A model trained on conversations generated by MIND from a small seed corpus can achieve math accuracy comparable to the DEEPSEEK MATH model trained on 120B unique tokens.	197
12.22	Training DEEPSEEK MATH-7B with Raw Data vs All Synthetic Dialogues. A strong pretrained LLM continuously trained on conversations generated by MIND provides significant boost in math accuracy than the same model trained on raw data—showing the effectiveness of MIND regardless of pre-training model quality.	198
12.23	Results of 7B LLM on Code Tasks: Conversations and rephrases are ineffective for code benchmarks.	199
12.24	Proportion of each dataset in different blends.	217
12.25	Token length statistics (Min, Max, Mean) for correct responses across task types.	218
12.26	Long-thought ablation (without tags): Impact of reasoning components on GSM8K and MATH-500 (math-hard). Configurations vary from question-to-solution only to multi-hypothesis with reflect and refine.	224
12.27	Long-thought ablation (with tags): Having multiple hypotheses, solutions and refinement phases in each solution yields the best math performances.	224
12.28	R1-style few-shot comparison: When few-shot prompts are formatted as long thought (R1-style), the long-thought-pretrained model (NuminaMath-ours: Q→3 Hyp→3 Sol→3 Ref) shows gains on math-hard and MATH-500 (R1 Style) over the baseline and over NuminaMath-only pretraining.	224
13.1	Breakdown of base model accuracies across benchmarks. With increasing diversity and quality, the difference between $\mathcal{M}_{\text{base}}$ and models pretrained with reasoning data increases.	226
13.2	Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{SHQ}	226
13.3	Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{LDQ}	227
13.4	Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{LMQ}	227
13.5	Results of all SFT models with varying pretraining and SFT data. Model pretrained with reasoning data obtains the highest gain after SFT phase of training.	228
13.6	Effect of early reasoning exposure on a 1.2B Transformer model. The reasoning-augmented model improves substantially on math, science, and code benchmarks while maintaining general-domain performance.	228

13.7	Impact of depth in reasoning traces in data on SFT phase. Model trained on longer CoT reasoning data outperforms the one trained on diverse reasoning traces.	229
13.8	Comparison between SFT with large amounts of unique data and reasoning-augmented pretraining. The reasoning-based model substantially outperforms the no-reason baseline despite using significantly fewer unique SFT samples.	231
13.9	Instruction-wise accuracy comparison on IFEval between models trained with less diverse (\mathcal{D}_{ALF}) and more diverse (\mathcal{D}_{LDQ}) corpora.	231
13.10	Effect of reducing scale and diversity of reasoning data during pretraining. \mathcal{M}_{ALF} is trained on a length-filtered subset of \mathcal{D}_{LDQ} that is smaller and less diverse.	232
14.1	List of hyperparameters used in both continuous pretraining (CPT) and finetuning (FT).	233
14.2	Results of DIFFERENCE-MASKING on multimodal video understanding benchmarks TVQA and Social IQ. DIFFERENCE-MASKING leads to an improvement of 8% and 2% accuracy over random accuracy.	234
14.3	We validate our hypothesis that masking patches using DIFFERENCE-MASKING is more effective than masking using CLIP similarity.	235
14.4	We validate our hypothesis that masking tokens using DIFFERENCE-MASKING at the word-level is more effective than masking at the token-level.	235
S.1	Ablations on rollout count, completion length, and KL weight β with QWEN3-1.7B-BASE. All numbers denote accuracy (%).	242
S.2	Ablation on math, science, and combined domains. RLP shows particularly strong generalization in presence of multi-domain data.	243
S.3	RLP training with QWEN3-14B-BASE model.	244
S.4	Comparison of NEMOTRON-NANO-12B-V2 4T Base and Base+RLP across benchmarks.	245
S.5	Comparison of CPT models with 8K vs 32K context length.	245
S.6	Effect of temperature τ on performance. Best result highlighted.	246
S.7	Comparison of SFT and RLP training efficiency.	246
S.8	Perplexity and NLL comparison on Nemotron CrossThink and Wikitext-103.	247
S.9	Quantitative benchmarks for QWEN3-1.7B-BASE, showing the impact of RLP on different data blends. Shaded columns indicate general pretraining corpus.	250

Chapter 1

Introduction

The ability to reason—logically deducing solutions, drawing inferences, and making decisions from available information—is a cornerstone of human cognition (Gendron et al., 2024). In recent years, Large Language Models (LLMs) have shown remarkable progress in performing both general reasoning and specialized tasks, including complex mathematical problem-solving (Hendrycks et al., 2021c; Cobbe et al., 2021a; Gemini, 2024; Nvidia et al., 2024; OpenAI, 2024b). Despite these advances, current models still struggle when faced with tasks requiring multi-step logical deductions, the ability to reflect on and correct mistakes, or to dynamically adapt their reasoning trajectory—especially in high-stakes domains such as mathematics, STEM, law, social sciences, commonsense, etc. (Glazer et al., 2024; Gao et al., 2024a; Hendrycks et al., 2021e)

In contrast to LLMs, human problem-solving is inherently iterative. We explore multiple reasoning paths, validate intermediate steps, and revise our approach based on feedback or errors (Kahneman, 2011). This recursive, self-reflective process builds toward more robust and accurate outcomes. However, LLMs are generally trained to produce linear outputs, often predicting a final answer without mechanisms to validate or improve upon intermediate reasoning. Even techniques like chain-of-thought prompting fail to fully replicate the rich, exploratory, and corrective nature of human reasoning (Huang et al., 2024a; Yang et al., 2023).

Improving the reasoning ability of LLMs thus requires not just better inference techniques, but a fundamental shift in the nature and structure of the pretraining data and the pretraining recipe. High-quality reasoning—particularly in mathematics—relies on structured, composite corpora that decompose complex problems into interpretable substeps and guide the model through logical chains of thought (Wei et al., 2022; Cobbe et al., 2021a; Gunasekar et al., 2023; Li et al., 2023b). However, curating such data is expensive, and access to the pretraining data of state-of-the-art models is limited, creating a barrier to reproducible progress in reasoning-focused research. Moreover, such reasoning enhanced data is even scarcer for other domains such as STEM, logical reasoning, law, social science, etc. These domains are inherently knowledge-centric, yet models with higher reasoning ability show better performance in such domains Team et al. (2025). This suggests that reasoning models are not only better at thinking and searching, but also good at probing their internal knowledge space — opening up a new direction on how to further enhance reasoning in underrepresented domains through pretraining.

To bridge this gap, synthetic data generation has emerged as a promising strategy to scale reasoning data effectively (Maini et al., 2024a; Gunasekar et al., 2023; Shah et al., 2024; Eldan & Li, 2023). Yet, most existing methods generate rephrased or static text that lacks the interactive, structured nature necessary for deep, multi-hop reasoning. They fail to simulate the back-and-forth exploration and explanation inherent in real-world problem solving. In addition, unlike post-training stages, the pretraining approaches do not explicitly reward the thinking ability of the model. Yet, recent works found that pretrained models have an implicit ability to recover from mistakes upon adding thinking (AI et al., 2025b) tags. This suggests that reasoning potential is already latent in pretrained models, but it requires the right inductive biases and supervision signals during pretraining to be fully realized.

In this thesis, we aim to understand the impact of structured reasoning data on pretraining Large Language Models (LLM), across a wide variety of reasoning domains that require multi-step and iterative logical reasoning. We plan to systematically observe the impact of such data in training across different phases of LLM development. In parallel, we intend to analyze the emerging thinking ability of the base model and find ways to incentivize good reasoning during pretraining. The subsequent sections of this chapter will outline the research background, followed by a concise introduction to the objective of this proposal and a detailed description of the proposed approach.

1.1 Background

The rapid advancement of Large Language Models (LLMs) has been predominantly fueled by scale—both in terms of model size and the volume of pretraining data. Scaling laws, such as those presented by (Hoffmann et al., 2022a), have established that optimal performance requires the simultaneous expansion of model parameters, data size, and compute. However, this ideal is increasingly challenged by limitations in data availability, cost, and quality.

1.1.1 The Importance and Challenges of Pretraining Data Curation

Large training datasets are an important driver of progress in the recent language modeling (LM) revolution (Gao et al., 2020a; Glorioso et al., 2024; Workshop et al., 2023; Soldaini et al., 2024). As the cost of training state-of-the-art language models continues to grow, researchers increasingly focus not only on scaling but also on improving training datasets that enable efficient generalization on a wide range of downstream tasks. To best benefit from scaling, recent scaling laws dictate that both model size and dataset size should jointly be increased (Hoffmann et al., 2022a).

Pretraining data curation has emerged as a critical component in the development of performant LLMs. The curation choices made when developing a pretraining dataset can have a huge impact on the downstream capabilities and performance of an LLM. As such, pretraining dataset curation strategies are often treated as closely guarded trade secrets. In fact, there are many popular “open” language models whose parameters are publicly available but whose pretraining datasets were not released and are scarcely documented or entirely proprietary (Jiang et al., 2023; Team et al., 2024; Touvron et al., 2023b), creating a growing gap between public and private capabilities in LLM development. The lack of transparency around data curation pipelines, such as those used in LLaMA, Mistral, or Gemma, limits reproducibility and hinders research into data-centric model improvements.

Web-scraped data, often from Common Crawl or similar sources, constitutes a significant portion of pretraining corpora. While web data is abundant, it is inherently noisy, unstructured, and frequently low in linguistic or factual quality (Eisenstein, 2013). Curation strategies such as filtering for quality, removing duplicates, and selecting high-value domains (e.g., educational, scientific, or encyclopedic content) have become essential (Touvron et al., 2023b; AI et al., 2025a; Shao et al., 2024). Yet, designing effective filtering pipelines remains a labor-intensive and largely heuristic-driven task, with limited consensus on best practices.

Moreover, the joint scaling paradigm of both data and model size raises significant challenges: although plentiful, text data is not infinite, especially so when considerations on data quality and licensing are taken into account—leading some researchers to argue scaling may soon be bottlenecked by data availability (Villalobos et al., 2024). Concretely, optimally training a GPT-3-sized model (175B parameters) would require no less than 3,500 billion tokens of text according to Hoffmann et al. (2022a). This is twice as much as the largest pretraining datasets ever demonstrated (Hoffmann et al., 2022a; Touvron et al., 2023b), and ten times more than the largest publicly available English datasets such as OSCAR (Suárez et al., 2019), C4 (Raffel et al., 2020a), or The Pile (Gao et al., 2020b).

Massively scaling up pretraining data is made even more challenging by the fact that LLMs are commonly trained using a mixture of web crawls and so-called “high-quality” data (Lozhkov et al., 2024; Li et al., 2024d; Zhihong Shao, 2024). Typical highquality corpora include curated sources of books, technical documents, human-selected web pages, or social media conversations. The increased diversity and quality brought by these curated corpora are believed to be a key component of performant models (Le Scao et al., 2022). Unfortunately, curation is labour-intensive: typically, each source requires specialized processing, while yielding a limited amount of data. Furthermore, scaling high-quality data is constrained by both legal and logistical issues. Licensed sources are limited in availability, and manually curated datasets are expensive to construct. This challenge is exacerbated by findings that repeated exposure to the same data across epochs yields diminishing returns and can induce overfitting (Muennighoff et al., 2023; Xue et al., 2023).

1.1.2 Synthetic Data for Augmenting Pretraining Quality

To address limitations in natural data, synthetic data generation has emerged as a promising solution that is relatively easier to obtain with more controllable quality (Bauer et al., 2024; Liu et al., 2024b; Long et al., 2024). Recent work has explored generating high-quality synthetic corpora using strong LLMs. For example, the WRAP (Maini et al., 2024a) approach demonstrates that rephrasing raw web text using off-the-shelf LLMs can improve data quality and enable better generalization in downstream tasks. Synthetic datasets such as Tiny Stories (Eldan & Li, 2023), Textbook-quality corpora (Gunasekar et al., 2023), and FineWeb (Penedo et al., 2024) have shown that small or mid-sized models trained on carefully generated synthetic data can rival or outperform models trained on larger, less curated corpora (Microsoft Research, 2023; Li et al., 2023b; Abdin et al., 2024b). Synthetic data for programming and math have also been adopted to improve the coding and reasoning abilities of LLMs (Guo et al., 2024a; Yu et al., 2024; Zhihong Shao, 2024). Previous studies have also focused on synthetic data for supervised fine-tuning (Eldan & Li, 2023; Zelikman et al., 2022; Huang et al., 2022; Liu et al., 2023a; Chen et al., 2024), instruction tuning (Wang et al.,

2022; Li et al., 2024b; Xu et al., 2023; Li et al., 2024c; Toshniwal et al., 2024b), downstream transferring (Meng et al., 2022; Ye et al., 2022), and evaluation (Zhu et al., 2023b; 2024).

Despite the wide usage of synthetic data, understanding what aspect of and how the synthetic data affect the performance of LLMs still remains largely unexplored, especially for pre-training. In the past, many studies have shown that both the quality and quantity of real data matter for LLM pretraining (Kaplan et al., 2020; Sorscher et al., 2022). Although synthetic data has shown promise in this paradigm, the synthesis process remains costly and often opaque, requiring millions or billions of API calls to proprietary models. In addition, synthetic data can introduce the risk of overfitting to specific tasks if generation prompts are narrowly scoped or biased (Maini, 2023). Existing studies on synthetic data in pre-training either only present methods of creating them (Maini et al., 2024a; Soldaini et al., 2024; Lozhkov et al., 2024) or provide findings that are restricted to relatively small scales (Eldan & Li, 2023; Microsoft Research, 2023), with limited understanding on what is the characteristic of high quality synthetic data and how exactly diversity of the synthetic tokens affect the training of LLMs.

To summarize, we identify the following core challenges in curating high-quality Pretraining corpora:

- **Limited transparency:** Pretraining data curation strategies are often undocumented, making it difficult to reproduce or improve upon state-of-the-art datasets.
- **Scarcity of high-quality data:** While crucial for downstream performance, high-quality data remains limited and expensive to obtain at scale.
- **Unclear structure for synthetic data:** Although synthetic data offers controllable quality, there is little understanding of how to structure or diversify it effectively across domains.

1.1.3 Emergence of Reasoning Ability in Pretraining

Recent studies indicate that large language models acquire non-trivial reasoning skills during the pretraining phase itself, even before any reinforcement learning (RL) or supervised fine-tuning (SFT) is applied (AI et al., 2025b). This finding challenges the common assumption that complex reasoning (e.g., multi-step problem solving with self-correction) only arises from post-training alignment or fine-tuning. Instead, evidence shows that standard next-token pretraining on vast data can already instill mechanisms analogous to those seen in RL- or SFT-tuned models.

AI et al. (2025b) demonstrates that pretrained models can perform self-reflective reasoning without any fine-tuning. In tasks with deliberately flawed intermediate reasoning (e.g., math, logic, coding), even base models can identify and correct errors when prompted. This ability strengthens with scale: larger checkpoints solve more adversarial tasks and require fewer reasoning steps. These findings suggest that self-correction and chain-of-thought reasoning are not artifacts of alignment or RL, but emerge naturally during pretraining.

Yue et al. (2025) explores whether RL with verifiable rewards (RLVR) enhances reasoning capabilities beyond the pretrained base. They find that RL-tuned models perform better at pass@1 (i.e., more efficient sampling) but do not generate fundamentally new solutions. Base models can often match or surpass RL-tuned models at higher sampling thresholds (e.g., pass@k). Thus, RL acts more as a distributional biasing mechanism—making correct

answers more likely—rather than expanding the reasoning space. In contrast, supervised distillation can introduce new problem-solving behaviors absent from the base model.

Together, these studies suggest that reasoning abilities such as multi-step inference, error detection, and revision are largely acquired during pretraining from patterns in natural text. Post-training techniques primarily enhance the expression and reliability of these capabilities rather than creating them. This reframes reasoning not as a post-hoc alignment outcome, but as an emergent property of large-scale pretraining. Further work may explore how to amplify this emergence via curriculum design, synthetic reasoning data, or early incorporation of alignment objectives.

1.2 Thesis Statement

The overarching goal of this thesis is to investigate the emergence and development of reasoning capabilities in Large Language Models (LLMs) during the **pretraining phase**, with a focus on understanding how structured reasoning data, inductive supervision signals, and multi-domain task formulations contribute to the formation of general-purpose cognitive abilities. While recent advances in post-training techniques—such as Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), and Direct Preference Optimization (DPO)—have demonstrated significant gains in alignment and response quality (Rafailov et al., 2023; Grattafiori et al., 2024), these approaches primarily refine capabilities that are already latent in the base model. In contrast, this work focuses on how such reasoning abilities originate during the earlier stages of model development and how they can be explicitly cultivated through carefully designed pretraining strategies.

This thesis pursues the following specific objectives:

- **Characterize the Role of High-Quality Reasoning Data in Pretraining:** We aim to identify the structural features of reasoning data—such as step-by-step chains of thought, recursive self-correction, and goal-conditioned problem solving—that enable pretrained models to perform complex, multi-hop reasoning. Drawing on recent findings that highlight the importance of educational value and structured trajectories in pretraining corpora (Wettig et al., 2024; Lozhkov et al., 2024; Li et al., 2024d), we will systematically assess how variations in data formulation (e.g., deductive proofs vs. Socratic dialogues vs. long-form solutions) affect the reasoning capabilities that emerge in LLMs.
- **Enable Generalization Across Non-Mathematical Reasoning Domains:** While most reasoning data and evaluation benchmarks are math-centric (Hendrycks et al., 2021; Glazer et al., 2024), real-world tasks in science, law, social science, and commonsense often require domain-specific patterns of inference and explanation. This thesis will extend reasoning-aware pretraining to *multi-domain contexts* by curating and integrating synthetic data from diverse fields (Akter et al., 2024; Yuan et al., 2025). Furthermore, we will explore scalable techniques for *verifiable reward modeling* in domains where correctness is not binary or deterministic¹, using templating and answer-space normalization to approximate feedback signals suitable for reinforcement-based supervision in pretraining.

¹Unlike mathematical domains with objective correctness criteria, general reasoning tasks such as commonsense inference often admit diverse yet plausible answers, requiring templating and normalization to build reward models.

-
- **Determine the Optimal Integration Phase for Reasoning Data:** Pretraining typically progresses through multiple stages, ranging from broad corpus ingestion to high-quality curriculum phases (Grattafiori et al., 2024; OLMo, 2025; Blakeney et al., 2024). A key objective is to empirically evaluate *when* in this pretraining pipeline, reasoning-focused data is most beneficial. We will study whether early exposure to structured logical tasks helps models internalize abstract cognitive patterns more robustly than late-stage augmentation, and whether such reasoning supervision should be constant or annealed over time. This analysis will be conducted across model checkpoints to track the evolution of inductive biases and representational shifts during training.
 - **Bridge Pretraining and Post-Training Paradigms via Hybrid Supervision:** Finally, we aim to investigate how post-training-inspired supervision techniques—such as lightweight RL-style feedback or preference modeling—can be incorporated *during* pretraining in a computationally efficient manner. This includes studying the effects of injecting “thinking cues” (e.g., reflective tags, reasoning prompts) (AI et al., 2025b) or distilling reasoning-aware policies into base models as part of the pretraining objective. The goal is to unify pretraining and post-training under a *reasoning-centric training pipeline*, enabling models to develop and express logical thought processes from the outset.

In summary, this thesis offers a principled exploration of **reasoning emergence in LLMs**, with a focus on systematically injecting and evaluating structured reasoning signals during pretraining. Through this investigation, we aim to derive generalizable insights into how models can be made inherently more interpretable, adaptable, and cognitively aligned—paving the way for LLMs that not only generate correct outputs, but reason their way to them.

1.3 Overview

In this section, we will briefly discuss the trajectories we want to explore throughout our thesis journey to address the objectives mentioned in Section 1.2.

Understand the Nature and Impact of High Quality Corpora. We start our exploration with the mathematics domain, as a strong mathematical reasoning ability highly relies on an abundance of high-quality, composite, and structured pretraining corpora. Curating complex, high-quality, structured mathematical data is costly and resource-intensive, largely due to the uneven distribution of high-quality sources. To mitigate this challenge, synthetic data generation has emerged as a scalable and cost-effective alternative. To generate high-quality data at scale, the current synthetic data generation approach explores rephrasing texts using LLMs in varied syntax while preserving the core content (Maini et al., 2024a). However, their proposed approach limits up-sampling high-quality data in a way that does not go beyond grammatical styles or surface form transformations—leading to little to no improvement when it comes to performance across complex and logical reasoning tasks. Therefore, we propose MIND (Akter et al., 2024), which transforms unstructured raw data into structured conversations—which simulates how people collaboratively explain, question, and reason—helping LLMs learn complex problem-solving skills. Compared to pretraining just on raw or rephrased data, a model pretrained on MIND-OWM shows a significant boost in mathematical reasoning, including superior performance in specialized knowledge and general-purpose reasoning tasks.

This work is evidence of the effectiveness of structured high-quality reasoning data in pretraining by prompting a stronger LLM in a zero-shot and simple single step. In this study, we want to formalize the structure of such reasoning data and how to curate or synthesize them. Inspired by MIND, we extend our work to multi-step prompting, where we define a format for ideal long thought data. In this step, we prompt the model sequentially, where the previous output is also fed to obtain the current thought. As mentioned in Section ??, we build a search tree by prompting LLM iteratively.

Ensuring General Improvement across Diverse Domains. Most recent works (Akter et al., 2024; Li et al., 2024d; Lozhkov et al., 2024) show a significant improvement in mathematical reasoning tasks, as structured and multi-hop reasoning chains are inherently beneficial for math reasoning. However, generalizing these gains to non-math domains—such as logic, scientific reasoning, commonsense inference, and law—remains a major challenge due to the unique nature of these tasks that relies on both reasoning and knowledge (Wang et al., 2024d; Team et al., 2025). To this end, we expand our investigation into the integration of long thought reasoning data across diverse domains. While reinforcement learning-based approaches such as DeepSeek-R1-Zero (Guo et al., 2025) show exceptional performance on math and code, their applicability remains limited outside these domains due to the difficulty of constructing verifiable reward functions. In this thesis, we address this gap by curating multi-domain synthetic reasoning data from raw web texts, employing templates and format normalization to constrain answer space variability and enable scalable reward modeling for general-purpose reasoning (Akter et al., 2025). We demonstrate that integrating multi-domain data improves generalization across benchmarks and enables reasoning-aware models to exhibit domain-agnostic cognitive behaviors. We plan to utilize these multi-domain aware reasoning models to curate more high-quality reasoning data across a wide range of domains, which will be injected during pretraining. Through these studies, we aim to identify whether reasoning-aware pretraining leads to cross-domain transferability or whether its benefits remain localized, ultimately guiding the construction of datasets and training objectives for broadly capable reasoning models.

Integrating Reasoning Data in Different Phases. Recent studies have gained a huge jump in downstream tasks by including high-quality data in post-training. However, Gandhi et al. (2025a) observes that the scale of improvement in post-training is dependent on certain capabilities of the base model. To be specific, base models that have strong reasoning ability are prone to achieving higher accuracy gain after post-training. In our work, integration of MIND (Akter et al., 2024) in pretraining has shown a significant boost in both mathematical and non-math reasoning tasks. As we have little to no details about the pretraining corpora of such reasoning models, we want to understand how the inclusion of reasoning data like MIND translates to the post-training paradigm. Our investigation will include two phases: (1) Does the base model pretrained on reasoning data provide a further boost in reasoning after post-training compared to the base model trained without the reasoning data? (2) Does the base model trained without reasoning data give an equal boost in downstream tasks after post-training when we include the reasoning data in the post-training? Both studies will provide insight into when and where this reasoning data is most beneficial.

Reasoning Incentive Pretraining. While most existing LLMs are pretrained using the next-token prediction objective, this approach does not explicitly reward models for engaging in

deep or structured reasoning. However, recent findings suggest that reasoning ability can be elicited from pretrained models through prompt-based self-reflection strategies (AI et al., 2025b). These observations open up a new direction for enhancing LLM cognition during pretraining by incorporating inductive signals that favor reflective and iterative thinking.

In this work, we aim to build a framework for *reasoning incentive pretraining*, where models are periodically guided to reflect on intermediate reasoning steps and self-correct through chain-of-thought prompts or reward-inspired inductive cues. We aim to understand whether lightweight mechanisms—such as inserting “think step-by-step” tags, reranking intermediate thoughts based on plausibility, or prompting for counterfactuals—can shift model behavior towards more deliberative problem-solving, even without explicit supervision or external reward. Our study will investigate the alignment between emergent reasoning behaviors and such incentive signals across pretraining checkpoints, providing insight into how early reasoning tendencies can be amplified, rather than solely extracted, during later alignment stages. Ultimately, this line of investigation seeks to unify pretraining and post-training techniques into a single, coherent reasoning-centric objective that endows LLMs with stronger intrinsic reasoning skills and generalizable cognitive scaffolds.

1.4 Chapter Overview

This thesis is organized around three central research questions: (1) understanding the anatomy of high-quality data for reasoning (Part I), (2) examining the effect of reasoning data across different stages of large language model development (Part II), and (3) redesigning the learning pipeline to enable the emergence of reasoning capabilities (Part III). Each part of the thesis contains chapters that investigate specific aspects of these questions. In the following chapters, we lay out the progression of this thesis through our recent works.

- **Part I: Anatomy of high quality data for reasoning**
 - **High-Quality Data for Generative Models (chapter 2):** This chapter lays the conceptual foundation for Part I by examining what distinguishes genuinely useful data for reasoning from data that merely adds volume. I survey the key attributes of high-quality reasoning data—including educational value, factual grounding, and structured inferential content—and situate them within the broader context of pretraining data curation and synthetic data generation. I then map out the four concrete problem settings addressed in the chapters that follow: visual reasoning, unimodal reasoning with multimodal models, structured data synthesis for pretraining efficiency, and generalization beyond mathematical domains.
 - **Data Synthesis for Improved Visual Reasoning (chapter 3):** This chapter addresses a fundamental flaw in existing visual question-answering benchmarks: the assumption that every question has a valid answer given the image. I introduce VISREAS, a large-scale dataset that explicitly includes unanswerable questions constructed from scene graphs, requiring models to verify question-image consistency before responding. I also propose LOGIC2VISION, a model that decomposes visual questions into pseudocode-style reasoning programs and executes them sequentially, without relying on external APIs or modular pipelines. Experiments show that existing VQA models are brittle under these conditions, while LOGIC2VISION demonstrates substantially stronger performance.

-
- **Unimodal Reasoning with Multimodal Models (chapter 4):** This chapter explores a self-supervised strategy for improving text-only reasoning by exploiting the visual capabilities of multimodal models. I introduce Self-Imagine, which prompts a Vision-Language Model to convert a text-based reasoning problem into an HTML-rendered visual representation and then reason over that self-generated image to produce an answer. The approach requires no additional data, supervision, or training. Experiments with LLaVA-1.5 and Gemini Pro confirm that self-generated visual scaffolds can meaningfully improve performance, and I analyze when visual representations help versus hurt, motivating future work on adaptive image generation.
 - **Structured Data Synthesis for Efficient Pretraining (chapter 5):** This chapter presents MIND (Math Informed syNthetic Dialogue), a framework for transforming raw web text into structured multi-turn dialogues that expose the step-by-step reasoning underlying complex problems. I show that existing rephrasing-based approaches lack the interactive structure necessary for deep multi-hop reasoning, and that conversations in which participants with different knowledge levels collaboratively work through content produce substantially richer training signals. MIND-OWM, the resulting 64-billion-token corpus, yields improvements of up to 6.29% on mathematical reasoning benchmarks and outperforms models trained on raw corpora 3.6 times larger, with gains extending to general-purpose reasoning tasks.
 - **Scaling Self-Learning beyond Math Reasoning (chapter 6):** This chapter extends reinforcement learning with verifiable rewards to domains beyond mathematics, where reward functions are difficult to construct due to diverse and non-deterministic answer spaces. I introduce NEMOTRON-CROSSTHINK, a framework that curates multi-domain data from sources spanning STEM, humanities, law, and social sciences; applies structured templates to constrain answer-space variability; and blends data sources using optimized ratios for GRPO-based self-learning. The best-performing configuration—using a 2:1 ratio of general-purpose to math data—achieves a 13.36% average improvement over strong baselines, demonstrating that data diversity, not data volume, is the primary driver of generalizable reasoning under the RLVR paradigm.
 - **Part II: Effect of reasoning data across LLM development phases**
 - **Synergy between Pretraining and Post-Training Data (chapter 7):** This chapter presents the first systematic study of how SFT-style chain-of-thought reasoning data interacts with different phases of the LLM training pipeline. I investigate whether a model pretrained without reasoning data can catch up through more intensive SFT, whether early exposure to reasoning data causes overfitting, and whether using the same data in both pretraining and SFT leads to mastery or forgetting. Experiments on an 8B model trained to one trillion tokens reveal an asymmetric allocation principle: diversity and scale drive pretraining effectiveness, while quality governs SFT. I show that front-loading reasoning data into pretraining creates durable foundations that post-training alone cannot recover, and that naive SFT scaling with noisy data can be actively harmful.
 - **Part III: Redesign the learning pipeline to enable reasoning emergence**

-
- **Bridging the Gap between Pretraining and Post-Training (chapter 8):** This chapter introduces Part III by framing the core problem: improvements from post-training are often fragile because models trained purely with next-token prediction have no incentive to form intermediate reasoning steps. I motivate two directions developed in subsequent chapters—difference-aware masking during continued pretraining, and reinforcement-based pretraining objectives—grounding both in recent findings that pretrained models already harbor latent reasoning capabilities that can be amplified by the right inductive signals.
 - **Difference-Masking: Choosing What to Mask in Continued Pretraining (chapter 9):** This chapter introduces DIFFERENCE-MASKING, a masking strategy for continued pretraining that automatically prioritizes tokens most distinctive to the target domain rather than selecting them at random. The key insight is that random masking wastes learning capacity on content the model already understands; by computing a TF-ICF score that measures how strongly each token characterizes the target domain relative to the pretraining distribution, DIFFERENCE-MASKING focuses the learning signal where it matters most. I evaluate the approach across four diverse language and multimodal video tasks, where it consistently outperforms random and attention-based masking baselines.
 - **Reinforcement as Pretraining Objective (chapter 10):** This chapter presents RLP (Reinforcement Learning Pre-training), which embeds reinforcement learning directly into the pretraining objective before any supervised fine-tuning has occurred. The core idea is to treat chain-of-thought generation as an exploratory action before each next-token prediction and reward it in proportion to the information gain it provides—specifically, the increase in log-likelihood of the observed token when the model conditions on its generated thought versus when it does not. This yields a verifier-free, dense reward signal applicable to ordinary web-scale text without task-specific answer checkers. I provide theoretical guarantees linking expected reward to reductions in cross-entropy, and show empirically that RLP consistently outperforms compute-matched baselines across datasets, domains, and architectures, with gains that persist and compound after downstream alignment.

Conclusion (Part IV): The final chapter synthesizes the thesis’s contributions around three central claims: that strong reasoning depends on carefully structured supervision rather than scale alone; that the placement of reasoning supervision across training phases profoundly shapes its durability; and that training objectives themselves can be redesigned to cultivate reasoning from the outset. I revisit each contribution, articulate the unified perspective that reasoning should be treated as a property cultivated throughout the entire learning pipeline, and outline future directions around reasoning trace anatomy, teacher model pipelines for non-mathematical domains, and tighter integration of pretraining and post-training through hybrid reinforcement objectives.

Part I

Anatomy of High Quality Data for Reasoning

Chapter 2

High Quality Data for Generative Models

The previous era of Large Language Model (LLM) development was defined by a singular obsession: *scale*. This was codified by the famous Chinchilla scaling laws (Hoffmann et al., 2022b), which formalized a now widely adopted principle: for a fixed compute budget, model size and training data must be scaled in equal proportion to achieve optimal performance. In particular, every doubling of parameters should be matched by a doubling of training tokens. This result reshaped the field. It suggested that, rather than allocating all resources to larger models, one could train smaller models on substantially more data and surpass larger but undertrained models.

At first glance, this appears to be an encouraging and tractable recipe for progress. If performance is compute-optimal at a particular data-to-parameter ratio, then increasing data should systematically unlock gains. In practice, however, this principle exposes a fundamental bottleneck. As we approach the "data wall," where high-quality human-generated text becomes a finite resource, the discourse has shifted. Scaling models is an engineering challenge; scaling reliable, diverse, and legally usable data is a structural constraint. It is no longer enough to simply ingest the entire internet; the frontier of AI research now lies in the curation of quality and the architecture of synthetic reasoning.

This tension reframes the central question. If scaling laws require more data for better performance, *what kind of data must be scaled?* When data becomes the bottleneck, quality and structure become first-order concerns. If the pretraining data is the "fuel" for these models, then web-scraped data is often crude oil—abundant but full of impurities that can lead to "unstable" behaviors, such as hallucinations or a failure to verify the premises of a user's query. The field increasingly recognizes that not all tokens contribute equally (Allal et al., 2025; Abdin et al., 2024a). A smaller corpus that is carefully curated or structurally enriched can outperform a much larger but unstructured alternative. In other words, once data scaling collides with data scarcity, the emphasis must shift from quantity to the anatomy of quality.

Tons of prior works have attempted to mitigate the challenge above. Previous works have explored these domains mostly by employing large language models to further refine the raw text, or improve the data quality by filtering out low quality data (Maini et al., 2024a;

Gunasekar et al., 2023; Yang et al., 2025b; Ishibashi et al., 2025). However, this provides a sudden boost in benchmark performance, but we are bounded by the knowledge of the teacher model. Plainly synthesizing data has diminishing returns beyond the optimal mix. With the risk of model collapse from recursive self-training. In addition, purely synthetic pretraining performs worse than mixed with real data (Kang et al., 2025; DatologyAI et al., 2025). Sometimes the model becomes extremely good at benchmarks but equally bad at general purpose reasoning tasks (Maini, 2023). This brings us to the conclusion that naively synthesizing data is not the solution to break the “Data Wall”!

This chapter explores the internal anatomy of what makes data “high quality” for reasoning tasks. We argue that high-quality data is not merely “clean” text, but data that forces a model to decompose problems, verify its own assumptions, and bridge the gap between different modes of representation (e.g., text, logic, and vision). Across three case studies, spanning multimodal reasoning, unimodal reasoning enhanced through visual scaffolding, and large-scale synthetic conversational pretraining, we demonstrate that reasoning performance depends less on raw token volume and more on structural properties embedded in the data.

2.1 Defining Attributes of High-Quality Data for Reasoning

Bringing together the empirical lessons from the preceding studies, we can now articulate a more complete definition of high-quality reasoning data, applicable to both pretraining and post-training settings. The central insight is that reasoning performance does not scale merely with token count, but with the structural and cognitive properties embedded within those tokens. High-quality data must therefore shape how models think, not only what they predict. We identify six core attributes that define the internal architecture of high-quality data, driving generalization in modern generative models (VLMs and LLMs):

Structured Reasoning Flow and Logical Sequence. Effective reasoning data does not present isolated question–answer pairs detached from their inferential process. Instead, it captures the logical sequence that connects premises to conclusions. This structure may be expressed through stepwise mathematical derivations, semantic programs aligned with visual scenes, or multi-turn analytical decompositions. What matters is that intermediate transformations are made explicit. By repeatedly observing coherent reasoning trajectories, models are encouraged to internalize compositional and multi-hop inference patterns. In contrast, unstructured corpora incentivize shortcut learning and pattern matching without deeper abstraction.

Conversational Dynamics as “Inner Monologue”. Human reasoning rarely unfolds as a single declarative statement; it involves questioning assumptions, refining partial ideas, revisiting earlier steps, and progressively clarifying the problem space. Data that encodes these dialogical dynamics provides supervision over the process of thinking itself. Structured conversations, role-based exchanges, or iterative refinements simulate this internal reasoning loop. Such formats strengthen robustness by exposing models to partial reasoning states and controlled revisions, rather than only polished final outputs.

Cognitive Diversity and Multiple Solution Paths. Reasoning tasks often admit multiple valid solution strategies, and high-quality data should reflect this plurality. When corpora

encode varied decomposition styles, alternative representations, and diverse explanatory pathways, models are less likely to overfit to rigid templates. Cognitive diversity enhances flexibility and supports adaptation to unseen problem structures. In synthetic pipelines, this diversity can be systematically encouraged by varying conversational roles, reasoning prompts, structural constraints, or representational formats. The result is a richer training signal that expands the model’s hypothesis space.

Generalization Power and Domain Transfer. Beyond internal diversity, high-quality reasoning data must exhibit generalization power. Its benefits should extend across domains and tasks. A dataset constructed for multimodal verification should improve reasoning robustness even in unimodal settings. Structured mathematical dialogues should enhance broader logical inference benchmarks. When improvements transfer beyond the original construction domain, this indicates that the data strengthens fundamental reasoning mechanisms rather than narrow task-specific heuristics. Generalization, therefore, becomes a defining empirical test of data quality.

Efficiency and Scalable Learning. Given the constraints of the Chinchilla scaling laws, data must be efficient and scalable. This means the data should have a high "information density" per token, allowing the model to achieve superior performance with fewer examples. By utilizing synthetic engines—such as those that generate millions of unique questions from scene graphs—we can accelerate the model’s learning curve, providing the "scale" required by modern compute budgets without the prohibitive cost of manual human curation.

Self-Expanding Potential. Finally, the powerful characteristic of data synthesis is self-expanding potential. Given the scarcity of curated high-quality corpora, the most sustainable datasets are those that can generate new structured examples from existing seeds. Synthetic conversational frameworks, programmatic transformations, and recombinable reasoning templates enable recursive growth. When a dataset can continuously produce diverse, structurally coherent reasoning instances, it transforms from a finite resource into a generative engine. This self-expansion property directly addresses the data bottleneck implied by compute-optimal scaling laws, offering a pathway toward sustained improvement without exclusive reliance on externally sourced corpora.

2.2 Problem Statements

2.2.1 Visual Reasoning

In visual question answering (VQA), validating question authenticity with the corresponding image and then reasoning over it is an important requirement in real-world application dynamics where users may make errors in judgment, leading to invalid queries. Confirming a question’s validity becomes pivotal to maintaining consistency, rectifying mistakes, and preventing misguided responses [Rajpurkar et al. \(2018\)](#). Following the prior VQA datasets’ [Goyal et al. \(2017\)](#); [Krishna et al. \(2016\)](#); [Hudson & Manning \(2019b\)](#) focus on answerable questions only, a system trained solely for answerable questions may exhibit unstable behaviors when faced with unanswerable queries. For instance, a delivery robot receiving an incorrect address but a valid instruction like “place the package by the yellow door”

might overlook the error unless prompted to reevaluate its decision. In contrast, presuming the correctness of the query would likely lead to unpredictable behaviors. Therefore, a reliable and responsible system should be able to question the validity of the instruction it receives before acting upon it.

However, aligning questions with the region of interest in the image breaks down the visual reasoning task into perception (object detection and scene representation learning) and reasoning (question interpretation and inference grounded in the scene). Datasets and models proposed to date have shown significant improvement in the detection task, which therefore improved the perception system Goyal et al. (2017); Krishna et al. (2016); Tan & Bansal (2019), but they face critical vulnerabilities due to the lack of generalities in the datasets Zhang et al. (2016b); Agrawal et al. (2016). Recent datasets Johnson et al. (2017); Selvaraju et al. (2020); Hudson & Manning (2019b) encourage reasoning beyond surface-level object recognition and focus on multi-step inference. But they tend to reason about object relations (*often questions revolving around a single object*) instead of reasoning over clusters of objects in the image that share common attributes or relations. Reasoning over general sets of objects requires both identifying objects and understanding their attributes and relations. Where prior scene-graph based work assumes reasoning follows from traversing a single path to generate an answer, our goal is to establish a multi-hop approach of identifying cliques with shared properties.

2.2.2 Unimodal Reasoning with Multimodal Models

Vision Language Models (VLM) are getting increasingly adept at solving a wide range of reasoning tasks (Liu et al., 2023b;d; You et al., 2023; Ye et al., 2023; Chen et al., 2023b; Zhang et al., 2023a; Chen et al., 2023a; Dai et al., 2023b; Lu et al., 2023). As these capabilities advance, VLMs are set to replace the current text-only language models for general-purpose interfaces like BARD (GoogleAI, 2023) and ChatGPT (OpenAI, 2021). In such scenarios, the deployed VLM would be required to handle a wide variety of end-user queries. Crucially, this includes queries that are not inherently multimodal, such as math-reasoning problems or program synthesis (Cobbe et al., 2021a).

A key question arises in these situations: How should a VLM, capable of functioning in a text-only mode like an LLM, handle text-based queries? While the default approach is to process these queries purely as text, this method does not fully exploit the VLM’s capabilities in image processing. Recent studies on human problem-solving provide a clue to addressing this gap: humans often draw visual representations to better understand and solve problems (Boonen et al., 2014; van Garderen et al., 2012; Krawec, 2014).

2.2.3 Structured Data Synthesis for Efficient Pretraining

The ability to reason is a fundamental element of human cognition, encompassing our ability to think logically, draw conclusions, and make decisions based on available information (Gendron et al., 2024). Large Language Models (LLMs) have demonstrated remarkable performance across wide range of general reasoning and specialized knowledge tasks. In particular, the improvement of LLMs in solving complex mathematical reasoning tasks (Hendrycks et al., 2021c; Cobbe et al., 2021a) has been significant in recent years (Gemini, 2024; Nvidia et al., 2024; OpenAI, 2024b).

Strong mathematical reasoning ability heavily relies on the abundance of high-quality, composite, and structured pretraining corpora. An effective mathematical corpus should not only contain relevant content but also be formatted to guide models to break down complex problems into smaller sub-problems and solve each part step-by-step—enhancing the model’s ability to process and reason about complex problems (Wei et al., 2022). Prior studies show that structured and well-formatted corpora play a crucial role in enhancing multi-hop and logical reasoning abilities (Cobbe et al., 2021a; Li et al., 2023b; Gunasekar et al., 2023), underscoring the importance of well-organized mathematical datasets in pretraining LLMs.

Curating complex, high-quality, structured mathematical data is costly and resource-intensive, largely due to the uneven distribution of high-quality sources. Most advanced models (OpenAI, 2024b; Gemini, 2024) are not publicly accessible, and it is unclear how their approach is enhancing math reasoning. To mitigate this challenge, synthetic data generation has emerged as a scalable and cost-effective alternative for creating a more balanced and diverse training corpus for pretraining LLMs (Maini et al., 2024a; Eldan & Li, 2023; Gunasekar et al., 2023; Shah et al., 2024). However, while these techniques have shown promise in improving general reasoning tasks, their data often lack the step-by-step problem solving structure crucial for multi-hop reasoning and complex mathematical tasks (Maini et al., 2024a), making them sub-optimal for such reasoning.

2.2.4 Generalize across Non-math Reasoning Domains

Long thought data has an evident impact on complex mathematical and logical reasoning tasks, as these tasks inherently require multi-hop reasoning. By deploying reinforcement learning on top of the base model, DeepSeek-R1-Zero Guo et al. (2025) has surpassed OpenAI’s o1-mini (OpenAI, 2024a) on math and code tasks. This model has later been used to generate cold-start data DeepSeek-R1’s SFT phase, which shows significant improvement across a wide range of tasks. This hints towards R1-Zero being the secret ingredient to build a reasoning-centric LLM. However, R1-Zero has been explored only on math and code domain as in these domains, building verifiable rewards are very straightforward. Extending this method to other reasoning domains such as logic, scientific reasoning, commonsense inference etc. is still an open question, as developing verifiable reward for domains where the answer space can be diverse is challenging. Synthetic data generation has emerged as a powerful technique for utilizing raw web texts, aiding both pretraining and post-training processes (Akter et al., 2024; Maini et al., 2024b; Ge et al., 2024). Synthetic question-answer pairs have significantly enhanced performance across various downstream tasks (Toshniwal et al., 2024a; Yuan et al., 2025), yet their utility in I-based training for non-math reasoning remains underexplored. A major challenge in applying I to general-purpose reasoning tasks lies in designing a verifiable reward model for diverse answer spaces, as, unlike mathematical reasoning—where correctness can be objectively verified—other reasoning tasks lack deterministic solutions.

In this work, we demonstrate a way to incorporate non-math reasoning domains in RL by utilizing raw web-texts. Integrating multi-domain data with different formats of questions for RL paves the way to maximize diversity in reasoning traces beyond a single domain, which leads to better generalization across diverse reasoning tasks. In addition, we address the challenge of designing verifiable rewards for non-deterministic domains by employing

different templates on the curated data to limit the nuances in the answer space diversity. This enables scalable, verifiable reward modeling for general-purpose reasoning tasks, ensuring that RL-trained models generalize effectively across diverse benchmarks. Furthermore, we show an optimal way to combine multi-domain data to ensure maximum gain across all reasoning tasks. We aim to investigate whether reasoning-aware pretraining leads to improved transferability across different tasks or if its benefits remain domain-specific. Additionally, we explore methods to create domain-agnostic reasoning datasets that allow LLMs to develop versatile reasoning capabilities applicable to a wide range of challenges.

Chapter 3

Data Synthesis for Improved Visual Reasoning

Verifying a question’s validity before answering is crucial in real-world applications, where users may provide imperfect instructions. In this scenario, an ideal model should address the discrepancies in the query and convey them to the users rather than generating the best possible answer. As discussed in Section 2.2.1, many existing datasets overlook this fundamental requirement and hence training with these corpora implicitly encourage models to produce an answer for every query, even when the question is invalid or cannot be resolved from the provided image.

3.1 Task Overview

Bridging the gap in prior benchmarks, we introduce a new dataset, **VISREAS** (**V**isual **R**easoning), for studying reasoning over commonalities and differences across objects. The unnatural assumption in the current VQA datasets - “*a correct answer for every question*” causes models to produce an answer even when the question is inapplicable and has no possible answer. To ensure that models verify the consistency of question text with the image before answering, we curate questions that have no answer given the image by altering relations and attributes among the objects. We design a question generation engine that takes the information about objects, attributes, and relations from the Visual Genome scene graphs [Krishna et al. \(2016\)](#) and finds common features shared among multiple objects. Based on this retrieved information, we generate 2.07M unique questions covering vast semantic variations. Each question is paired with a scene graph and a semantic program that specifies the series of reasoning steps needed to be performed to produce the answer. Our generated questions require visual reasoning abilities such as comparing, differentiating, counting, clustering objects, and performing logical reasoning. Most importantly, unlike other VQA datasets, **VISREAS** enforces the VQA models to verify the information in the question with the image in each reasoning step before predicting an answer.

We find existing VQA models less robust in the reasoning and unanswerable settings presented by **VISREAS**. Motivated by the shortcomings of existing models, we propose a new architecture, **LOGIC2VISION** that has been trained to produce logical reasoning steps from the query at first and then predict answers based on the reasoning steps and the image.

Unlike prior generative models, LOGIC2VISION is compute and cost-efficient as it does not require any external expensive APIs or modules and solely relies on the reasoning capabilities of visual language models (VLM).

In short, our contributions are twofold:

- We introduce **VISREAS**, a dataset containing complex yet natural reasoning. Our dataset makes the first step towards developing reliable VLM adaptable to real-world scenarios where user instructions may not always be impeccable.
- We present **LOGIC2VISION**, that aims to handle spatial reasoning by executing consecutive pseudocode with verification in each step.

3.2 Data Generation

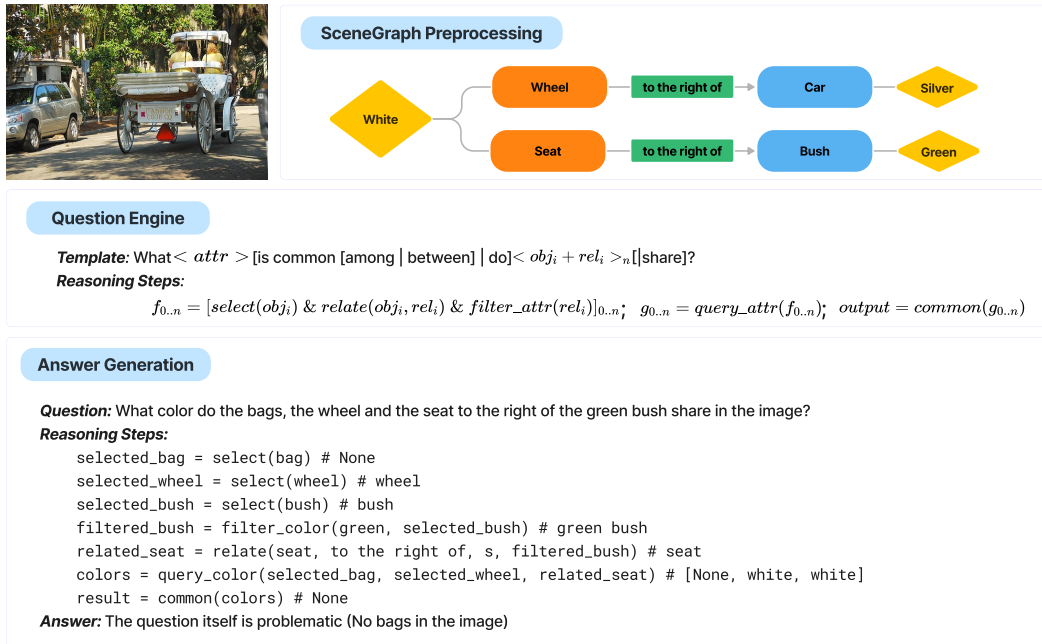


Figure 3.1: Overview of VISREAS dataset construction process. Using scene graphs, we cluster objects (orange), relations (green), and attributes of the related objects (blue) based on the attribute of the corresponding objects (orange). Then the question engine takes each template as input and traverses all possible clusters to generate the query as well as the reasoning steps. Each function in the reasoning steps can return **NONE** if any object, attribute, or relation is absent in the image.

The dataset is constructed in three major steps: (1) Process scene graphs, (2) Define templates and reasoning functions that the question will involve, (3) Automatically generate corresponding reasoning steps in pseudocodes along with the final answer from each query as shown in Fig. 3.1. Finally, to prevent models from learning statistical biases in attribute, reasoning, or answer type distributions, we meticulously balance the VISREAS dataset across three distinct paradigms (section 12.1).

3.2.1 Scene Graph Processing

To begin with the data construction process, we run two phases of processing on the scene graphs before passing them to the question engine.

First Phase. We clean up the scene graphs by removing opposite attributes and discarding object nodes with similar names that share similar attributes and relations. Our processed scene graphs contain 1703 distinct objects, 14 attributes, and 114 relationships. It is also observed that one object name in the image might correspond to multiple object IDs and bounding boxes in the scene graph. This will cause ambiguity in the later question-generation process. Thus, we merge bounding boxes corresponding to the same object name with a high IoU (> 0.7). In addition, there can be images where a bigger bounding box contains multiple small bounding boxes, which can be either parts of the object represented by the bigger bounding box (e.g., a cat (bigger bounding box) has a tail, ear, face (small bounding boxes), etc.) or they can collectively represent the object in the bigger bounding box (e.g., lime and apple can together be mentioned as fruits). These overlapping bounding boxes will be problematic while clustering objects based on similar attributes (e.g., fruits and lime are all green; for ‘*What has the same color as the lime?*’ the answer generation module will produce: fruits and apple - which is ambiguous). To discard these cases, we measure the ratio of intersection area vs individual bounding box area and check whether the smaller objects are subclasses of the bigger one using Wordnet Miller (1994). If the ratio is high and the larger object is a superclass of the smaller one, we discard the larger bounding box during preprocessing to avoid ambiguity.

Second Phase. We cluster the scene graphs based on the common attributes and relations among the objects in each image and create several sub-graphs as seeds for the question engine. Initially, we cluster objects based on a single relation or attribute, later we merge the clusters recursively if there are objects with multiple attributes or relations in common. Finally, each cluster represents a collection of objects that share a similar set of attributes and relations and the question engine exhaustively traverses all clusters to generate questions. For each object in a cluster, we also store other objects that are related to that object along with their relation name. This information is used to populate nested compositional references for multi-hop relation traversal.

Question Engine

For question generation from the clusters, we manually create 182 templates on different attributes (Table 3.1). Our templates cover five categories of reasoning (*query*, *count*, *compare*, *verify*, and *choose*) which can be further broken down into nine broad categories of reasoning mentioned in Appendix. For some categories, we have list answers and no-answer cases. All of our templates are formulated considering clusters of objects to facilitate multi-object comparison. To generate no-answer cases, we apply two approaches: (1) We either add an outlier (object not present in the image) to the cluster or include an object that exists in the image but not in the cluster and has different relations and attributes from the objects in the cluster. (2) We perturb the existing relation/attribute of an object inside a cluster (e.g., change ‘*apple to the left of knife*’ to ‘*apple to the right of knife*’) which derives no-answer cases.

Attribute	Templates	Train	Validation
Color	12	1326086	1500
Cleanliness	8	7794	900
Material	15	368337	1500
Size	4	116438	1500
Pose	18	36687	1500
Height	10	9894	1200
Weather	6	31376	1500
Length	11	45764	1500
Tone	11	37184	1500
Shape	15	30119	1500
Activity	21	15639	1500
Sport Activity	21	13215	1500
Age	12	19594	1500
Pattern	18	14313	1500
Total	182	2072440	20100

Table 3.1: Question-template distribution over attributes

Answer Generation

The answer generation step involves two consecutive phases. *Initially*, we formulate the reasoning steps in pseudocode (Figure 3.1) and produce the intermediate results for each line of code using our designed parser (Figure 3.2). For each question template and reasoning type, we have hand-coded the basic reasoning steps necessary to answer the query. Based on the number of objects, relations, and attributes, our parser generates all intermediate reasoning steps along with the answers. *Finally*, we combine all intermediate results to come up with the answer. If any intermediate reasoning step results in ‘NONE’, the final answer becomes ‘the question itself is problematic’ indicating some objects, relations, or attributes mentioned in the question text cannot be found in the image.

3.2.2 Dataset Analysis and Comparison

The VISREAS dataset consists of 113K images from the Visual Genome where each image is annotated with dense descriptions of the scene stored in the scene graphs. We refine the existing scene graphs and generate 2,072,437M unique questions, twice the size of current VQA datasets (Table 3.1), that combine features of multiple objects and their relations and require the implementation of consecutive complex reasoning skills with an in-depth understanding of object attributes and relations in the image. Our dataset covers 14 different attributes and 114 diverse relations among 1703 different objects from real-life images. We define five major types of reasoning (Figure 3.3) while generating the corpus based on the overall nature of the query template. Figure 12.1 shows details of the query structures along with examples. However, the intermediate reasoning steps that are necessary to answer the query can be diverse and can combine all five types of reasoning for a single query (as in Figure 3.1). We combined the data set over 14 attributes and 5 reasoning types (section 12.1).

Compared to existing VQA tasks, VISREAS emphasizes creating longer reasoning chains (multi-hop) with a larger number of objects (Figure 3.3). The average number of reasoning hops for VISREAS is 1.42 (95% CI: [1.415, 1.417]), significantly higher than GQA (mean: 0.52; 95% CI: [0.517, 0.519]) and CLEVR (mean: 0.84; 95% CI: [0.839, 0.843]). However, to limit the

```

# question: contains question, answer, and semantic string
# scene: collection of objects in the scene
def parse(question, scene):
    codes, outputs = [], []
    for line in question["semantic"]:
        code, output = run(line, scene, history)
        codes.append(code), outputs.append(output)
    return codes, outputs

# line: single line of semantic string
# scene: collection of objects in the scene
# history: list of previous outputs
def run(line, scene, history):
    # formats semantic string to pseudocode, operator, and its arguments using regex
    # IN "select:box(43)" OUT "selected_box = select("box")", "select", { id:"43" }
    pseudocode, operator, args = parse_semantic_string(line)
    match operator:
        case "select":
            return scene.objects[arg.id]
        case "relate":
            { relation, subject, object } = args
            objects = scene.objects.find(o => o.relations[subject.id] == relation)
            return objects
    # implement other operators
    case "...":
        ...

```

Figure 3.2: **Semantic string parser.** For every line of semantic string, we use regex and string manipulation to extract operator and its arguments. We represent scene-graph in adjacency list format and run the parsed operator to get formatted pseudocode and its expected output.

question length and increase human readability (Figure 12.2), the majority of the questions require at most two hops relation traversal for each object.

Reflecting on human clustering ability based on commonalities, consists of queries that require consideration of multiple objects based on their attribute or relation similarities. Therefore, unlike existing datasets, the majority of queries are composed of more than three objects from the image. The average objects per question for is 3.91, which is higher than both GQA (1.12) and CLEVR (1.63). Hence, requires multiple object detection and consecutive reasoning to answer a single query (Figure 3.3). In addition, each query can have multiple attributes associated with it (Figure 12.3a). For example, in question, ‘What is the common material among the silver and blue utensils?’, both <material> and <color> attributes are needed to be considered for answer generation that involves multiple attribute filtering along with the associated objects.

In contrast to other spatial reasoning datasets that focus primarily on one-hop relation traversals (Bottom left of Figure 3.3), we explore two ways of novel traversals: (1) *Star Relation*: The target object shares multiple relations with other objects (e.g. is the center of

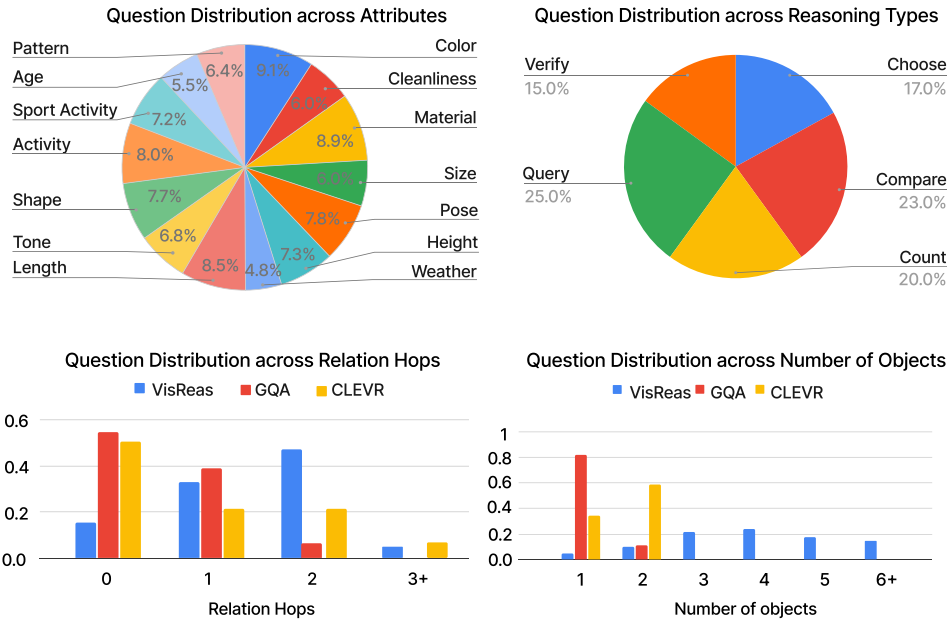


Figure 3.3: Overview of VISREAS statistics. **(Top left)** The dataset covers 14 attributes in a balanced ratio. **(Top right)** It consists of five reasoning types of queries in a balanced distribution. **(Bottom left)** Comparison of multi-hop relation traversal for different VQA datasets. Majority questions of VISREAS require multi-hop traversal compared to others. **(Bottom right)** Comparison of number of objects mentioned in the question for different datasets where VISREAS questions contain larger amount of objects.

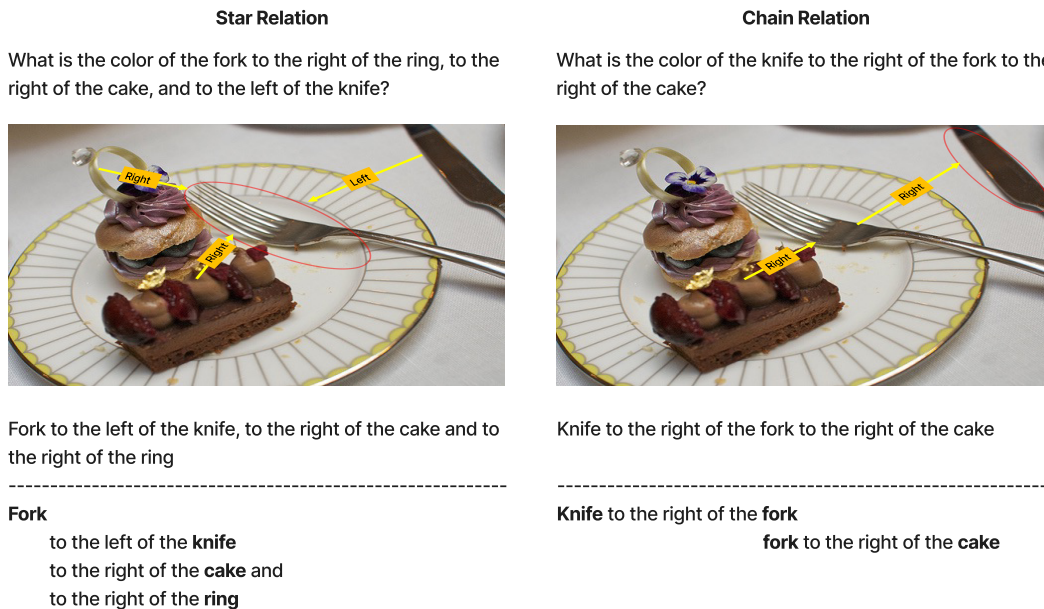


Figure 3.4: VISREAS contains two types of relation traversals. **Star relation** states a single object that shares multiple relations with other objects (Left). **Chain relation** states multiple objects that share a single relation with each other (Right).

the star and other objects are connected to it with a relation – [Figure 3.4 left](#)), and (2) *Chain Relation*: The target object is related to an object that is related to another object and the relation traversal is linear ([Figure 3.4 right](#)). The inclusion of these traversals adds multi-hop complexity to the corpus and makes the *each-step verification process* harder for unanswerable questions (as [Figure 12.5](#)).

3.3 Visual Reasoning with Structured Data

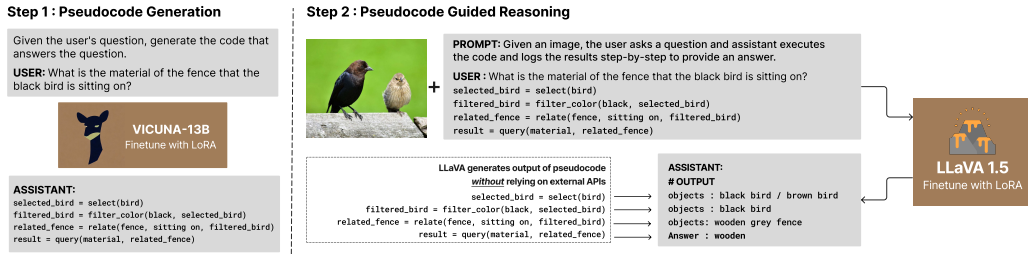


Figure 3.5: Overview of LOGIC2VISION. In **Pseudocode Generation** phase, we generate pseudocode which outlines the reasoning steps. During **Pseudocode-Guided Reasoning**, the pseudocodes along with the question and image are provided to the model. The model executes all intermediate pseudocodes to arrive at the final answer.

In recent years, LLMs combined with code generation and chain-of-thought prompting have shown impressive performance in complex reasoning by generating intermediate reasoning steps before inferring the answer [Zhang et al. \(2023c\)](#); [Surís et al. \(2023\)](#). However, these frameworks are often prone to hallucinations of LLMs and are too restricted in terms of reasoning they can perform and dependent on expensive external modules to execute the reasoning [Zhang et al. \(2023b\)](#); [Surís et al. \(2023\)](#). To address these limitations and elicit the reasoning capability of VLMs, we propose , a two-stage VQA framework that (1) plans the necessary reasoning steps using the question and (2) executes the plan with the help of an image leveraging the SOTA VLM ([Figure 3.5](#)).

3.3.1 Pseudocode Generation

Given a natural language question, this module generates a consecutive set of reasoning steps as pseudocodes. For training our pseudocode generation model, we take advantage of the existing VQA dataset: GQA as it provides a semantic string that decomposes the question into a sequence of reasoning steps. For instance, the semantic string for the question ‘Is there a red apple on the table?’ would be ‘select: table → relate: on, subject, apple → exist: ?’. We build a custom parser ([Figure 3.2](#)) that converts each line of GQA semantic string to pseudocode and extracts all the intermediate expected outputs along with the final answer from the scene graph. The parsed (pseudocode, output) pairs serve as a rationale to solve the question ([Figure 12.4](#)). For the pseudocode generation, we use an instruction finetuned VICUNA-13B [Chiang et al. \(2023\)](#) model which has shown good performance across various language tasks including code generation. We finetune VICUNA using LoRA on (question, pseudocode) pairs [Hu et al. \(2022\)](#) to generate the pseudocode for a given question. The finetuned model achieves 98.6% METEOR [Banerjee & Lavie \(2005\)](#)

score and 96.3% ROGUE-L [Lin \(2004\)](#) score against ground-truth code parsed from GQA semantic strings.

3.3.2 Pseudocode-Guided Reasoning

Since the Pseudocode Generation module outlines the necessary steps to answer the question, the remaining task is to perform pseudocode-guided sequential reasoning on the image. For this stage, we choose state-of-the-art VLM, LLaVA-1.5 [Liu et al. \(2023b\)](#), due to its impressive performance in diverse reasoning tasks. As LLaVA-1.5 was not trained to reason with pseudocode and image, we fine-tuned it to generate an answer by executing sequential reasoning with the pseudocode and the image. To adapt this framework in our case, we rearrange the instruction as below:

```
USER:<Image> Executes the code and logs the results step-by-step to provide an answer to the question.
Question: {Question}
Code: {Code}
ASSISTANT:
Logs: {Logs}
Answer: {Answer}
```

Here LOGIC2VISION takes the image, question, and the corresponding sequential pseudocodes as input and produces all intermediate outputs of the codes as logs along with the final answer. Therefore, during fine-tuning, not only learns to generate the final answer but also must predict all intermediate responses correctly. This includes predicting NONE when there is no answer possible in any intermediate step. The ability to produce intermediate outputs as logs makes more explainable compared to others. As each line of the pseudocode requires a different reasoning ability (e.g., select, compare or relate), we can detect which reasoning task the model is failing by simply tracking the logs. The essential training details of this stage can be found in [section 12.3](#).

3.4 Experiments and Analysis

In the subsequent sections, we conduct a comprehensive analysis of the VISREAS dataset and assess the performance of various benchmarks including LOGIC2VISION, GPT-4V [OpenAI \(2023\)](#), and human participants, revealing a notable disparity from human performance.

3.4.1 Baseline Experiments

To analyze the complexity and generalizability of our dataset and model, we run experiments with models trained on both classification and generative tasks. We cover two types of generative models: **GEN** (relies on pretrained visual-language alignment module) and **Code-GEN** (generates a program and utilizes external APIs to solve VQA tasks). We categorize LOGIC2VISION as **Logic-GEN** as it produces intermediate logical reasoning steps before answering. All model configurations can be found in [section 12.3](#). To make the training and inference consistent, we define our own prompt for all generative models (as [section 12.3](#)). [Table 3.2](#) shows the results of different baselines on both GQA and VISREAS. All baseline models perform worse on VISREAS than on GQA, highlighting the unique

Metric	CLS			GEN			Code-GEN		Logic-GEN	GPT-4V	Humans
	LXMERT	ViLBERT	CRF	BLIP-2	InstructBLIP	LLaVA-1.5	ViperGPT	VisProg	LOGIC2VISION		
Choose	74.23	82.91	83.30	71.21	78.50	84.11	10.37	15.86	82.54	82.61	91.30
Compare	65.62	69.86	71.87	28.72	53.29	67.75	5.97	26.09	59.25	68.33	86.12
Count	45.32	47.80	49.59	25.88	49.86	43.08	7.85	6.02	39.47	39.52	85.78
Query	44.05	47.65	48.11	41.55	47.77	50.31	4.35	19.30	63.79	58.78	81.78
Verify	76.10	82.18	83.03	70.77	49.48	81.27	3.10	44.18	84.54	82.16	93.94
Problematic	67.54	77.08	78.41	25.39	64.68	68.04	0.25	0.16	55.34	70.18	90.29
Non-Problematic	56.11	59.16	61.60	51.41	52.25	60.31	11.97	24.17	67.94	55.47	84.89
Accuracy (%)	50.15	53.05	53.56	47.81	53.97	61.38	10.38	20.82	66.20	62.83	87.21

Table 3.3: Accuracy breakdown of baseline models and humans on VISREAS across different reasoning types. **Problematic** questions contain relations, attributes, or objects that are missing or inconsistent with the image, whereas **Non-Problematic** questions are consistent with the image. Except for the **Code-GEN** models, we report fine-tuned results on VISREAS.

challenge provided by VISREAS. Table 3.3 presents the performance on VISREAS across diverse baselines along with GPT-4V and human accuracy. We break down the performance along two axes: the reasoning type and answerability. We finetune models in the **CLS** and the **GEN** groups to obtain stronger baseline results. We could not finetune models in the **Code-GEN** group due to their close-sourced weights. **Logic-GEN** outperforms all others baselines at a significant margin.

[CLS] For models trained with **classification task**, we finetune and evaluate on both GQA and . From the fine-tuning results of the CLS models, it is obvious that proposes a different task than GQA that can not be easily solved by scaling the model size or changing the pretraining corpus. Furthermore, the higher performance gap of the models between GQA and tasks suggests the inefficacy of the existing CLS models on our proposed spatial reasoning task.

	Model	Accuracy (%)	
		GQA	VISREAS
ZS	BLIP-2 (Li et al., 2023a)	44.70	35.16
	GEN InstructBLIP (Dai et al., 2023a)	49.50	36.84
	LLaVA-1.5 (Liu et al., 2023b)	63.3*	38.98
	Code-GEN ViperGPT (Suris et al., 2023)	48.10	10.31
	VisProg (Gupta & Kembhavi, 2023)	50.50	20.82
FT	LXMERT (Tan & Bansal, 2019)	60.05	50.15
	CLS ViLBERT (Lu et al., 2020)	60.65	53.05
	CRF (Nguyen et al., 2022)	72.10	53.56
	Logic-GEN LOGIC2VISION	60.32	66.20

Table 3.2: Performance comparison among baseline models on GQA and VISREAS. (*) GQA trainset images were used during training.

[GEN] From **generative domain**, we select three SOTA models, BLIP-2, InstructBLIP, and LLaVA-1.5, that try to leverage the LLMs using two types of vision-language alignment modules: Q-Former and MLP cross-modal connector. We evaluate the models on zero-shot GQA and VISREAS to probe the relevance of our proposed task to their training domain. We notice that BLIP-2 performs poorly on our task compared to GQA where InstructBLIP and LLaVA-1.5 shows higher accuracy. Both LLaVA-1.5 and InstructBLIP are instruction tuned on diverse downstream tasks which allows them to excel in VQA tasks compared to BLIP-2. However, LLaVA-1.5 gains the highest zero-shot accuracy in this category due to its training set images being overlapped with VISREAS. Yet, it shows a significant drop (-24.32%) in ZS accuracy compared to GQA, which proves that highlights a novel reasoning task that can not be generalized using GQA. Furthermore, the smaller performance gap among these models on VISREAS suggests the inefficacy of the current VLMs on our proposed spatial reasoning task.

[Code-GEN] From modular Code Generation models, we analyze recent works - ViperGPT and VisProg. These models employ an LLM to generate an executable program that utilizes a pre-defined API, including functions such as `detect(image, obj_category)` or `segment(image, obj_category)`. VisProg also utilizes the “in-context learning” abilities of LLMs, enabling the model to respond to new queries with just a few examples of input and output behavior. Zero-shot evaluations of Code-GEN models on GQA and VISREAS reveal that current models are struggling with our task more than GQA, where both corpora use similar images. We find these models heavily biased to answerable setting that they tend to ignore the discrepancies between the question and the image. Furthermore, the codes generated by these models are often incomplete or runs into error when passed to the compiler. We term these cases as incorrect responses for consistent evaluation. We believe that problematic questions can be handled better with modified prompts which would require additional expensive few-shot prompting. However, their poor performance in *Non-Problematic* questions denotes the inability of these models to reason with longer relational hops and cluster multiple objects based on commonalities.

3.4.2 Analysis

According to Table 3.3, all the models including GPT-4V struggle in Compare, Count, and Query question-types which require grounding, clustering, and verifying the existence of multiple objects, relations, and attributes. Specifically in Query, the performance gap between humans and the models is significantly higher which demonstrates the limitation of current models to perform complex multi-hop reasoning. , on the other hand, shows a promising result in Query questions. We hypothesize that structured pseudocode helps the model consider each object and its corresponding attributes and relations before answering while the other models try to learn from the surface-level word distribution. In addition, Query questions are in general lengthier than other types of questions which makes it easier for the models to lose attention to the details (Figure 12.3b).

In contrast, GPT-4V outperforms all generative models in Problematic questions. After analyzing the predictions, we find that GPT-4V excels at identifying problematic questions that involve an object not present in the image or an object with a false attribute. However, when the question becomes problematic due to an incorrect relation, GPT-4V consistently struggles to recognize it which also holds for other models. This signifies the uniqueness of our corpus that emphasizes understanding relations beyond simple object detection. It is also notable that GPT-4V often denies to answer questions related to a person and sometimes just ignores questions by saying ‘I’m sorry, but I can’t assist with identifying or making assumptions about people in images.’ For fair comparison with other models, we report all these occurrences as incorrect answers.

Model	Choose	Compare	Count	Query	Verify	Prob.	Non-Prob.	All
7B	81.20	54.90	35.13	59.24	82.75	55.38	63.92	62.74
13B	82.54	59.25	39.47	63.79	84.45	55.34	67.94	66.20

Table 3.4: Breakdown of accuracies on VISREAS for ‘s VICUNA model size. We observe that VICUNA’s model size improves performance in most question-types except the problematic ones.

To investigate the effect of LLM’s scale on the VQA task, we test two versions of LLMs (VICUNA 7B and 13B) within architecture. [Table 3.4](#) breaks down the performance of in the presence of different LLMs. We observe that increasing LLM’s size dramatically increases the accuracy of longer questions ([Figure 12.3b](#)) such as Non-Problematic, Count, Query, and Compare instances and marginally improves performance on question categories such as Choose and Verify. This finding reassures the ability of larger LLM to reason with longer context. However, for problematic questions, increasing LLM size has no impact. As this category requires verification and grounding of information with image, both LLM and vision-language alignment need to be strong to excel in this domain.

3.5 Related Works

Recent years have witnessed tremendous progress in visual understanding. Multiple attempts have been made to mitigate the systematic biases of VQA datasets [Goyal et al. \(2017\)](#); [Zhang et al. \(2016c\)](#); [Agrawal et al. \(2018\)](#); [Johnson et al. \(2017\)](#), but they fall short in providing an adequate solution: Some approaches operate over constrained and synthetic images [Zhang et al. \(2016c\)](#); [Johnson et al. \(2017\)](#), neglecting the realism and diversity natural photos provide. [Suhr et al. \(2019\)](#) introduced a dataset for reasoning about semantically-diverse natural language descriptions of images in the form of a classification task. While the dataset exhibits diverse semantic phenomena, this task rarely requires much beyond a single type of object recognition and its associated relation and attribute. Unlike these datasets, is open-ended and consists of both unanswerable and answerable queries based on the similarity/dissimilarity of multiple objects in the image. jointly evaluates VQA models’ alignment, multihop reasoning, and verification ability which cannot be approximated by simply finding the most likely object/relation/attribute to answer the question.

Recent transformer-based models have [Tan & Bansal \(2019\)](#); [Lu et al. \(2020\)](#); [Nguyen et al. \(2022\)](#) achieved promising performance on visual reasoning tasks. Yet, these models are prone to reproducing spurious correlations without accurately learning true causal relations [Agrawal et al. \(2016\)](#); [Jia & Liang \(2017\)](#); [Tenenbaum \(2018\)](#). Neural-symbolic methods [Andreas et al. \(2016\)](#); [Hu et al. \(2017\)](#); [Hudson & Manning \(2018; 2019a\)](#) explicitly perform symbolic reasoning on the object and language representations. These models offer modularity and interpretability in the reasoning process. However, as module parameters are usually derived solely from end-task supervision, there is a potential for the program to deviate from accurately explaining the model’s behavior [Ross et al. \(2017\)](#); [Jain & Wallace \(2019\)](#); [Subramanian et al. \(2020\)](#).

Conversely, a recent approach to modularity leverages Large Language Models (LLM) to craft code or Python programs using expensive APIs [Chen et al. \(2021b\)](#); [Surís et al. \(2023\)](#); [Gupta & Kembhavi \(2023\)](#); [Subramanian et al. \(2023\)](#). However, these approaches outsource basic aspects of the reasoning to external components rather than performing reasoning as part of the model itself. For example, prior works outsource basic cognitive abilities such as recognizing objects, counting, and even arithmetic operations. Focusing on these limitations, our proposed aims to leverage single VLM to address complex reasoning in a modular approach that shows promising performance across models of three different categories.

3.6 Summary

In this chapter, we examined how synthesizing structured, high-quality data can address a fundamental limitation of existing visual language systems: their difficulty in performing complex spatial reasoning while verifying whether a question is grounded in the visual context. Many current VQA benchmarks emphasize answer prediction under the assumption that every question is valid and answerable. This design choice encourages models to rely on superficial correlations or object recognition patterns rather than performing structured reasoning about relationships among objects in the scene. As a result, even state-of-the-art models often fail when faced with compositional queries, multi-hop reasoning steps, or questions that are inconsistent with the visual evidence.

To address this gap, we introduce the dataset, for real-world complex and multihop visual reasoning and compositional question answering. The dataset emphasizes object commonalities, differences, and relational aspects, necessitating validation of question-text relevance with the image before answering. We describe the dataset curation process along with the performance of SOTA models from three different domains in our task. Addressing the shortcomings in grounding and clustering in recent models, we propose a novel baseline that deconstructs questions into pseudocodes and sequentially executes them using images to generate answers. We anticipate that this dataset and model will catalyze advancements in VQA research, pushing it toward complex semantic comprehension, robust reasoning, and addressing unanswerability when the provided context is not sufficient.

Solving VQA tasks via code generation and external APIs has gained attention due to its capability to perform complex reasoning and planning in a modular manner. However, code generation has limitations: a fixed set of operations limits models to specific types of questions and heavy use of external modules prevents end-to-end training. While modularity encourages specialization, in practice it requires managing multiple environments and heavy GPU memory usage as multiple large models are used to carry out visual and cognitive tasks like detection and captioning. In addition, current code generation methods [Surís et al. \(2023\)](#); [Gupta & Kembhavi \(2023\)](#) rely on OpenAI’s API to generate executable code which hinders the accessibility of benchmarking due to its high costs¹ and fluctuations of OpenAI models over time [Chen et al. \(2023c\)](#) which makes it hard to diagnose whether certain performance gains come from OpenAI model or improvements in other components. In contrast, our model and dataset suggest that one can use a single VLM model that combines both the strength of structured reasoning and train it in a simple end-to-end manner. requires many operations such as *select*, *filter*, *relate*, and *query* which are limited to cognitive skills to standard VQA tasks and spatial reasoning. Therefore, models trained on may not generalize well for visual-language tasks such as visual storytelling and image captioning which goes beyond the scope of our dataset. A natural future direction would be to incorporate other visual-language tasks into the dataset as well.

¹Evaluation with VisProg requires approximately 2,500 tokens per question including in-context examples, prompts, and outputs. Using original `text-davinci-003` model used in original code would cost $(0.0200/1000 \text{ tokens}) \cdot 2500 \text{ tokens} \cdot 17171 \text{ instances} \approx 858 \text{ USD}$.

Chapter 4

Unimodal Reasoning with Multimodal Models

Vision Language Models (VLM) are getting increasingly adept at solving a wide range of reasoning tasks (Liu et al., 2023b;d; You et al., 2023; Ye et al., 2023; Chen et al., 2023b; Zhang et al., 2023a; Chen et al., 2023a; Dai et al., 2023b; Lu et al., 2023). As these capabilities advance, VLMs are set to replace the current text-only language models for general-purpose interfaces like BARD (GoogleAI, 2023) and ChatGPT (OpenAI, 2021). In such scenarios, the deployed VLM would be required to handle a wide variety of end-user queries. Crucially, this includes queries that are not inherently multimodal, such as math-reasoning problems or program synthesis (Cobbe et al., 2021a).

A key question arises in these situations: How should a VLM, capable of functioning in a text-only mode like a Language Language Model (LLM), handle text-based queries? While the default approach is to process these queries purely as text, this method does not fully exploit the VLM’s capabilities in image processing. Recent studies on human problem-solving provide a clue to addressing this gap: humans often draw visual representations to better understand and solve problems (Boonen et al., 2014; van Garderen et al., 2012; Krawec, 2014).

4.1 Task Overview

Building on the insight of human thought process, we propose SELF-IMAGINE—a technique designed to enhance the reasoning abilities of VLMS on text-only tasks through visualization (Figure 4.1). initially generates a graphical representation of the text query using the VLM. Then, the *same* VLM is used to solve the problem using both the original question and the self-generated image.

An inherent challenge is that advanced VLMS are not typically equipped for direct image generation. To circumvent this, we utilize the VLM’s code generation capabilities to generate HTML code visually representing the query information. This HTML is then rendered as an image, which, when used in conjunction with the original text query, allows the VLM to operate with both textual and visual information. With , the VLM efficiently serves dual purposes: generating visual representations and solving the problem. This strategy

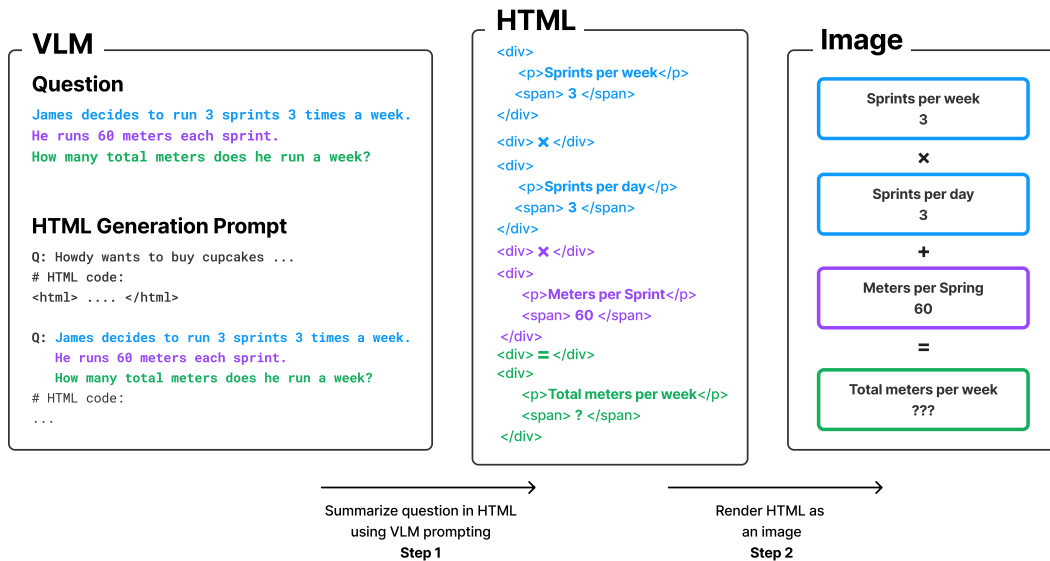


Figure 4.1: Generating an image from a question via a single VLM through HTML.

effectively reduces reliance on separate image generation models such as DALL-E (Shi et al., 2020), streamlining the problem-solving process.

4.2 Methodology

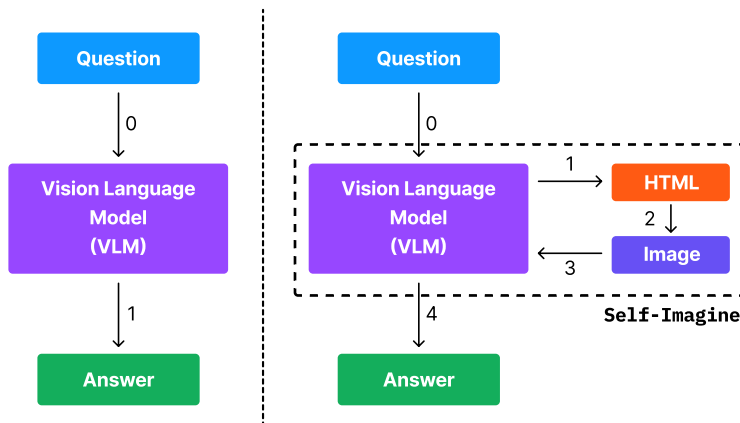


Figure 4.2: **[Left] Reasoning using VLM without** : Given a question (0), the VLM generates an answer (1). **[Right] Reasoning using VLM with** : Given a question (0), the VLM generates a structured representation of the question using HTML (1). The HTML is rendered as an image (2) which is then passed along with the question to the VLM again (3). The VLM finally generates the answer by combining both vision and language modalities (4).

Unlike Large Language Models (LLM), Vision Language Models (VLM) can combine multiple modalities in the same representation space and perform complex reasoning. However, when it comes to unimodal downstream tasks (e.g., math-reasoning), VLMs are not fully leveraged due to the absence of additional modalities. In , we circumvent this by generating a visual representation for a given reasoning question using the VLM in the form of an

image. Then, the same VLM is fed both the question and the generated image to answer the question. In the following section, we expand on the image generation from the question.

4.2.1 Generate Image from Question

While VLM cannot generate images directly, they are pre-trained on large corpus of programs and thus are proficient in code generation. Thus, we utilize code generation capabilities of these models to create an image for the question. While there are several choices for choosing a representation (SVG (St.Laurent et al., 2001), Tikz (Tantau, 2022)), we use HTML due to its prevalence and its ability to easily generate structured information from questions using tables, lists, flow charts, etc.

Generate HTML from Question. To convert natural language questions into HTML, we choose two Vision Language Models (VLM): LLaVA-1.5 (Liu et al., 2023b) & GEMINI PRO (Team, 2023), due to their impressive performance on a wide range of reasoning tasks. Since multimodal models are not traditionally trained for HTML generation, we approach this using a few-shot prompt, interleaving natural language questions with HTML codes. For each natural language question q_i , we generate a corresponding HTML code h_i . These are paired as $\langle q_i, h_i \rangle$ to form a prompt $p = \{q_j, h_j\}_{j=1}^K$, where $K = 5$ represents the number of in-context examples chosen for diversity in reasoning tasks. Given a new question q_t , we combine it with the prompt p and a placeholder image I_d , and input these into the VLM to generate the HTML h_t for q_t as shown in Equation 4.1.

$$h_t = \text{VLM}(p \parallel q_t, I_d) \tag{4.1}$$

Convert HTML to Image. After generating HTML from questions, we use the ‘imgkit’ python library to render these HTML codes into images. To evaluate the role of images in reasoning tasks, we conduct experiments both with and without the generated images. We append task-specific prompts to the questions, as detailed in Table 12.5. In the image-inclusive experiments, we use the HTML-generated images alongside the concatenated prompts and questions, inputting these into the VLM for processing.

$$\begin{aligned} I_g &= f(h_t) \\ y_t &= \text{VLM}(p \parallel q_t, I_g) \end{aligned} \tag{4.2}$$

Here, f represents the HTML renderer, and I_g represents the final generated image from the question. y_t is the answer generated using the question with the prompt $(p \parallel q_t)$ and the image (I_g).

4.3 Experiments

4.3.1 Experimental Setup

Benchmarks. We explore two kinds of reasoning tasks to evaluate our approach: (1) *math word problems* consisting of GSM8K (Cobbe et al., 2021a), ASDIV (Miao et al., 2020), and SVAMP (Patel et al., 2021) and (2) *symbolic reasoning* consisting of NAVIGATE, GEOMETRIC SHAPES, TRACKING SHUFFLED OBJECTS, PENGUINS IN A TABLE, COLORED OBJECTS, DATE

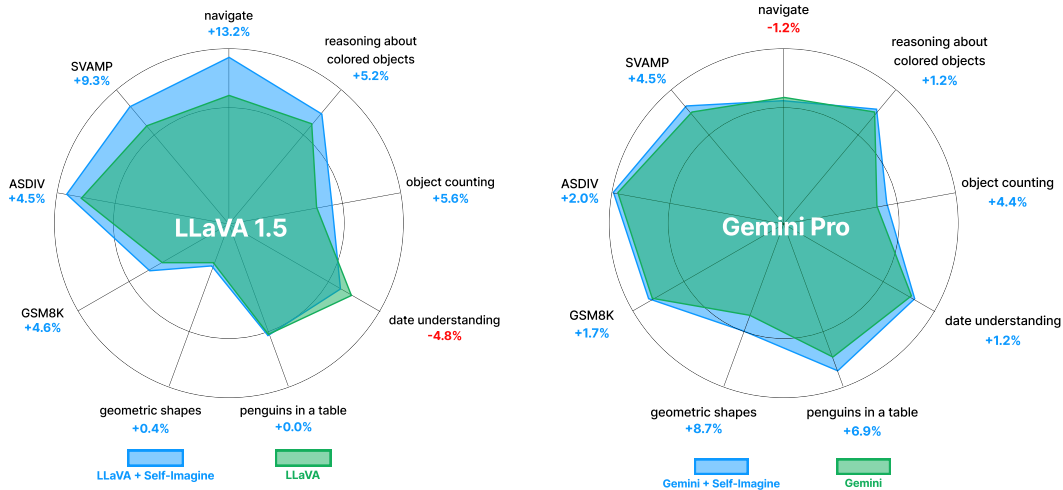


Figure 4.3: **SELF-IMAGINE main results:** SELF-IMAGINE improves accuracy over a diverse range of mathematical and symbolic reasoning tasks.

UNDERSTANDING, and OBJECT COUNTING tasks from BIG-Bench Hard (Suzgun et al., 2022).

Baselines. For the baseline, we consider zero-shot prompting where we only pass a basic prompt (Table 12.5) and the question. We performed greedy decoding from the language model using a temperature of 0. Note that this is a realistic setup for current open-source multimodal LLMs, which cannot accept a prompt interleaved with text and images.

Vision Language Models. We use LLaVA-1.5 (Liu et al., 2023b) and GEMINI PRO (Team, 2023) as our VLMS and keep each one of them consistent throughout the HTML generation phase to the question-answering phase. LLaVA-1.5 uses a CLIP ViT-L (Radford et al., 2021a) as a vision encoder and Vicuna 13B (Chiang et al., 2023) as the LLM. Conversely, GEMINI PRO is built on Transformer architecture (Vaswani et al., 2017a) and is trained with a wide range of multimodal data. The architecture of this model has not been disclosed yet. In this paper, we accessed GEMINI PRO through Google AI Studio. GEMINI PRO comes with default safety features that block certain questions, especially those involving potentially illegal or sensitive content. For our analysis, we disabled these safety settings.

Evaluation During the evaluation, we slightly modified the standard evaluation protocol (Gao et al., 2024b), which consisted of matching the words “The answer is” followed by a numerical output. We found that the VLM sometimes fails to follow this sentence verbatim even when it produces the correct answer. To accommodate these cases, we simply take the last number/option of the generated text as the answer to the question.

4.3.2 Results

We summarize our results across three math and nine reasoning tasks in Table 4.1. We define the baseline setup as ‘Question Only’ when we only feed the question with the basic prompt to the VLM. is indicated by the ‘Question + Image’ setup where we generate the HTML from

the question at first and pass the rendered image from HTML along with the basic prompt and question to the VLM as input (Equation 4.2).

improves the VLMs’ performance in all math reasoning tasks: for example, SELF-IMAGINE improves the base LLAVA-1.5 and GEMINI PRO by 9.30% and 4.50% accordingly in SVAMP. In OBJECT COUNTING (LLAVA-1.5: +5.60%; GEMINI PRO: +4.40%), COLORED OBJECTS (LLAVA-1.5: +5.20%; GEMINI PRO: +1.20%) and GEOMETRIC SHAPES (LLAVA-1.5: +0.40%; GEMINI PRO: +8.70%), inclusion of SELF-IMAGINE improves both VLMs.

LLAVA-1.5 and GEMINI PRO have different subsets of symbolic reasoning tasks in which they benefit from SELF-IMAGINE. In particular, LLAVA-1.5 benefits from SELF-IMAGINE in tasks involving multiple variables e.g., navigation and tracking multiple objects tasks, as the image provides additional structured information on top of the question. On the contrary, GEMINI PRO + SELF-IMAGINE excels in list and tabular reasoning tasks such as DATE UNDERSTANDING (+1.20%) and PENGUINS IN A TABLE (+6.85%). All these tasks require diverse reasoning abilities, and the improvement across these tasks represents the generalizability of SELF-IMAGINE.

However, hurts the performance of VLMs in some of the symbolic reasoning tasks - for LLAVA-1.5: DATE UNDERSTANDING (-4.80%) and TRACKING SHUFFLED OBJECTS of three objects (-2.80%); for GEMINI PRO: NAVIGATE (-1.20%), TRACKING SHUFFLED OBJECTS of three objects (-13.6%), of five objects (-13.6%), and seven objects (-5.60%). These tasks are easier to solve using only the question rather than having an image. The reason behind degradation with an image is two-fold: (1) the generated images are incorrect and visually not informative given the question (DATE UNDERSTANDING, NAVIGATE), (2) HTML cannot visually portray terms like swap between objects and cannot keep track of an object after multiple swaps (TRACKING SHUFFLED OBJECTS). These results indicate that stronger image generation capabilities that capture consecutive progression of reasoning might help to boost the performance of the VLM.

In the following section, we demonstrate that the improvement is highly correlated with the quality of the generated image, underscoring the dependency on the ease of converting text into an image. In addition, an image that appropriately captures the flow of reasoning always guides the VLM to the correct reasoning path.

Task	Dataset	LLAVA-1.5		GEMINI PRO	
		Question Only	Question + Image	Question Only	Question + Image
Math Reasoning	GSM8K	26.69	31.31 (+4.62)	74.37	76.04 (+1.67)
	ASDIV	52.24	56.73 (+4.49)	82.01	84.02 (+2.01)
	SVAMP	43.50	52.80 (+9.30)	69.50	74.00 (+4.50)
Symbolic Reasoning	OBJECT COUNTING	31.20	36.80 (+5.60)	46.40	50.80 (+4.40)
	NAVIGATE	44.80	58.00 (+13.2)	60.80	59.60 (-1.20)
	COLORED OBJECTS	44.80	50.00 (+5.20)	70.40	71.60 (+1.20)
	DATE UNDERSTANDING	50.00	45.20 (-4.80)	72.80	74.00 (+1.20)
	PENGUINS IN A TABLE	41.10	41.10 (0.00)	70.55	77.40 (+6.85)
	GEOMETRIC SHAPES	14.40	14.80 (+0.40)	48.00	56.70 (+8.70)
	TRACKING SHUFFLED OBJECTS (3)	33.60	30.80 (-2.80)	60.00	46.40 (-13.6)
	TRACKING SHUFFLED OBJECTS (5)	18.00	18.40 (+0.40)	41.20	27.60 (-13.6)
TRACKING SHUFFLED OBJECTS (7)	12.00	16.80 (+4.80)	34.40	28.80 (-5.60)	

Table 4.1: Comparison of accuracy between ‘Question Only’ and ‘Question + Image’ across diverse reasoning tasks where the image has been generated using SELF-IMAGINE.

4.4 Ablations

4.4.1 Math Reasoning

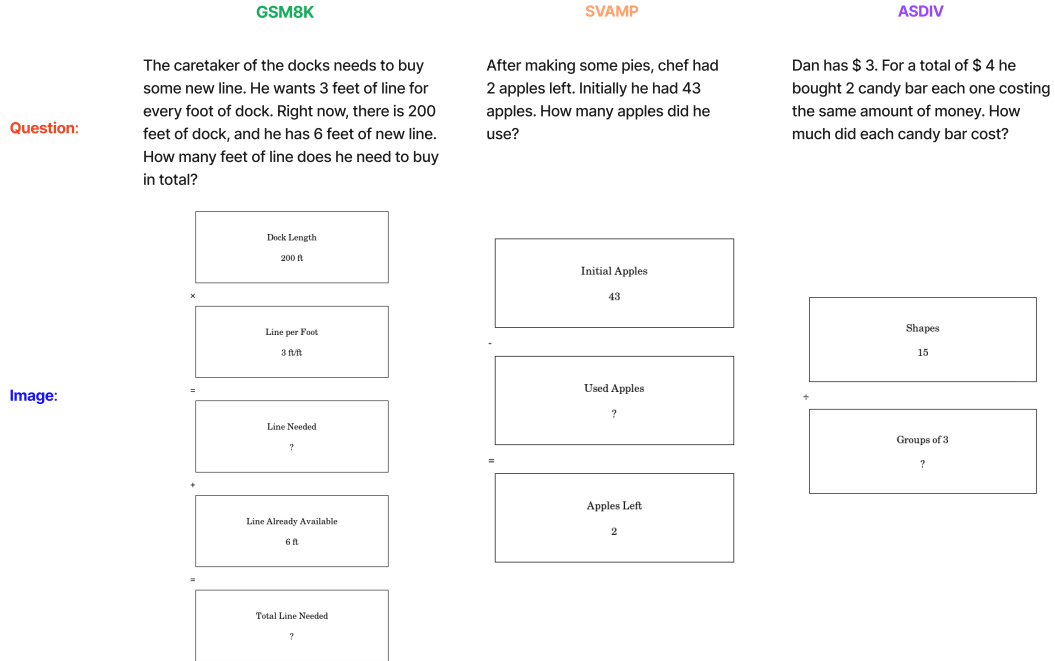


Figure 4.4: Example from math world problem tasks.

For math reasoning tasks, we analyze the performance of VLMS with and without image support. This analysis includes examining performance variations across question complexity, the length of the reasoning chain, and specific instances where images contribute positively or negatively to problem-solving. The generated images, as depicted in Figure 4.4, predominantly feature boxes, each labeling a variable and its value, designed to simplify and clarify the information presented in the question.

Why does image help? The primary advantage of using images lies in their ability to distill complex information into a more manageable format. In several tasks, particularly those involving substantial irrelevant data (e.g., GSM8K, ASDIV), an image serves as a focused reference point, enabling the model to concentrate on key variables and their values (see Table 4.2, Table 12.6 for examples). Additionally, images often include variable names marked with question marks, as shown in Figure 4.4, which guide the model in identifying the critical elements necessary for multi-step reasoning.

Image helps solve moderately complex questions. In general, longer questions tend to be complex. Here, we examine the performance variation regarding question length as detailed in Figure 12.10. We find that image helps LLAVA-1.5 more than GEMINI PRO in longer and more complex questions in ASDIV and SVAMP tasks. This finding aligns with the previous explanation, i.e., the image removes unnecessary verbose from the question, making the reasoning process easier.

Question	Stephen placed an online order for groceries. His final bill came to \$40.00. Because this was through a delivery vendor, they tacked on a 25% fee to his final total and charged him \$3.00 in delivery fees. Stephen also added a \$4.00 tip. After the extra fees, what was the final price of Stephen’s groceries?
Image	<pre> graph TD A["Total \$40.00"] --> B["25% Fee ?"] B --> C["Delivery Fee \$3.00"] C --> D["Tip \$4.00"] D --> E["Final Price ?"] </pre>
Label	57
Output With Image	<p>To find the final price of Stephen’s groceries, we need to calculate the total cost after the extra fees and tip have been added.</p> <ol style="list-style-type: none"> 1. Calculate the 25% fee: $\\$40.00$ (final bill) $\times 0.25 = \\$10.00$ (25% fee) 2. Add the delivery fee to the final bill: $\\$40.00$ (final bill) $+ \\$10.00$ (25% fee) $+ \\$3.00$ (delivery fee) $= \\$53.00$ (total cost with fees) 3. Add the tip to the total cost: $\\$53.00$ (total cost with fees) $+ \\$4.00$ (tip) $= \\$57.00$ (final price of groceries) <p>The answer is \$57.00.</p>
Output Without Image	<p>The final price of Stephen’s groceries was $\\$40.00 + 25\% \text{ fee} + \\3.00 delivery fee $+ \\$4.00$ tip $= \\$40.00 + 0.25 \times 40 + 3 + 4 = \\$40.00 + 10 + 3 + 4 = \\$40.00 + 13 = \\53.00. The answer is \$53.00.</p>

Table 4.2: Example of Image improving reasoning in GSM8K task for LLaVA-1.5.

However, we can also observe that for more complex questions in the GSM8K task (question length > 70 for LLaVA-1.5 & question length > 50 for GEMINI PRO), performance with images deteriorates compared to performance without images. This decline stems from the inadequate HTML generated by longer questions, which often fail to encapsulate all the necessary information. Therefore, images generated from those HTMLs confuse the VLMs rather than help.

This observation also holds for questions with longer reasoning chains depicted in [Figure 12.11](#) for the GSM8K task. Questions that require a longer chain-of-thoughts (COT) are not better represented with images for LLaVA-1.5. However, GEMINI PRO is robust to increasing COT length and rather benefits from having a structured representation for complex questions. This analysis also presents an opportunity for future research. It suggests that the most challenging questions, which intuitively could benefit the most from the structural and contextual support provided by images, are precisely where current methodologies for image generation fall short.

Why does the image hurt? While images generally enhance the VLM’s reasoning, specific scenarios lead to diminished performance. A notable issue arises during HTML generation, where the VLM occasionally pre-solves arithmetic sequences, embedding them into the image ([Table 12.7](#)). This can mislead the model if the embedded calculations are incorrect. Furthermore, certain concepts like ‘trade/exchange’ or ‘add/delete’ are challenging to represent visually, leading to inaccuracies in questions involving these terms. Another

complication involves questions with fractions, such as ‘Shelly ate $3/4$ of the eggs from a dozen.’ The corresponding images often depict these fractions in a simplified form (e.g., a box labeled ‘Already ate: $3/4 \times 12$ ’), which the model struggles to compute accurately as it requires the execution of multiple operations (i.e., division and multiplication) simultaneously. Similarly, when the VLM tries to execute multiple operations mentioned in the image, it usually generates the incorrect answer. For example, in [Table 12.11](#), with the image, the VLM executes four operations in a single line (i.e., $10 * 1/2 + 15 * 1/3$) and ends up generating the wrong answer. But without an image, the calculation is broken down even further, producing the correct answer. This problem might be solved with an improved image that breaks down each step as a single operation consisting of two numbers.

4.4.2 Symbolic Reasoning

In this category, we focus on nine diverse reasoning tasks from BIG-Bench Hard benchmark ([Suzgun et al., 2022](#)) to observe the importance of image. We break down the overall accuracy by tasks and analyze the performance by question complexity and answer types. The images generated with HTML for the tasks are images with labeled/colored boxes ([Figure 4.5b](#)), tables ([Figure 4.5a](#), [Figure 4.5c](#)). Occasionally, we find that the generated image simply contains the text (as in [Table 12.12](#)).

Why and when does image help? The overall accuracy indicates a decent improvement for LLaVA-1.5 (2.56%) with SELF-IMAGINE (as [Figure 4.3](#)) where GEMINI PRO receives slight accuracy loss (-1.69%) with self-generated image. We further break down the results across the tasks. As shown in [Figure 4.3](#), adding an image augments the performance of LLaVA-1.5 in the majority of symbolic reasoning tasks while achieving comparable performance in others. In parallel, adding images improves GEMINI PRO in tasks that require shape, color, list, and tabular reasoning such as COLORED OBJECTS, OBJECT COUNTING, DATE UNDERSTANDING, PENGUINS IN A TABLE, and GEOMETRIC SHAPES.

For COLORED OBJECTS, PENGUINS IN A TABLE, and OBJECT COUNTING tasks, the VLMs generate well-structured tables or multiple boxes in rows with variable names in one column and corresponding values in another column. Thus, when solving with an image, the reasoning problem simplifies to finding column sums or specific table elements. Notably, GEMINI PRO, being a decent table parser ([Akter et al., 2023](#)), excels in these tasks with images. In GEOMETRIC SHAPES, the HTML simply depicts the shape provided in the SVG vector. As a result, image helps both VLMs by providing a visual reference of the intended shape in the question (as [Table 12.10](#)).

In contrast, TRACKING SHUFFLED OBJECTS requires tracking multiple objects through consecutive swaps. As mentioned in [subsection 4.4.1](#), VLMs struggle to depict swaps between objects using HTML ([Table 12.8](#)), leading to degradation in performance with the image in TRACKING SHUFFLED OBJECTS tasks. However, TRACKING SHUFFLED OBJECTS of five and seven objects tasks are notably more complex than the three objects task as they require tracking more objects and their swaps and text-only LLaVA-1.5 performs poorly in these tasks. Having an image that logs each object attribute and their swaps, helps LLaVA-1.5 to solve the question accurately rather than having no image. Hence, we can see the improvement of LLaVA-1.5 with the increasing number of objects for TRACKING SHUFFLED OBJECTS tasks.

Question: I have a table, and three cars. How many objects do I have?

Image:

Objects Count	
Table	1
Cars	3
Total Objects	?

(a) OBJECT COUNTING

Question: On the floor, you see a bunch of objects arranged in a row: a fuchsia jug, a black teddy bear, a blue envelope, a silver paperclip, a teal pencil, and a yellow keychain. What is the color of the object furthest from the silver object?

Image:



(b) COLORED OBJECTS

Question: Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. We now add a penguin to the table: James, 12, 90, 12 How many penguins are less than 8 years old and weight more than 12 kg?

Image:

Name	Age	Height (cm)	Weight (kg)
Louis	7	50	11
Bernard	5	80	13
Vincent	9	60	11
Gwen	8	70	15
James	12	90	12

(c) PENGUINS IN A TABLE

Figure 4.5: Examples from some BIG-Bench Hard sub-tasks.

Finally, in NAVIGATE task, LLAVA-1.5 significantly improves with image inclusion, while GEMINI PRO shows little degradation in accuracy. Unlike other tasks, the NAVIGATE task is challenging to depict using HTML. Therefore, most of the images generated with both VLMS for this task contain texts either showing the question or necessary reasoning steps in natural language (Table 12.12). Without an image, LLAVA-1.5 performs poorly compared to GEMINI PRO on this task. However, with images, the LLAVA-1.5 executes additional reasoning during HTML generation, thereby increasing the likelihood of predicting the correct answer in the presence of an image. This phenomenon also explains GEMINI PRO’s improvement in the DATE UNDERSTANDING task with images, as the generated HTML primarily offers reasoning steps in natural language.

Image helps with shorter (GEMINI PRO) and more complex questions (LLAVA-1.5). Following subsection 4.4.1, we investigate the impact of the image in the reasoning process with increasing question length. Here, we observe distinct behaviors in two VLMS. As depicted in Figure 12.12, LLAVA-1.5 benefits from images with both simpler, shorter questions and more complex ones, while GEMINI PRO’s performance declines as question length increases. Generating high-quality HTML is also easier for simpler and shorter questions, which benefits

both VLMs during question answering with the appropriate image. However, with longer questions, the generated HTMLs tend to ignore some information or can not summarize all information in a structured manner. This results in lower performance compared to without image setup. Interestingly, [Figure 12.12](#) shows higher performance for LLAVA-1.5 in with-image setup when the question length exceeds 120 words. This category includes the TRACKING SHUFFLED OBJECTS of seven objects task, which requires tracking seven objects during multiple swaps. As mentioned earlier, the image depicts the role of each object in the question and their swaps, which simplifies the reasoning process. Unlike LLAVA-1.5, GEMINI PRO particularly tries to solve the swaps step-by-step in the HTML rather than just logging the information. However, it fails to keep track of the objects after multiple swaps - resulting in the generation of an incorrect HTML and a dramatic drop in accuracy with images.

Why does the image hurt? Despite the benefits observed in certain tasks, incorporating images into the reasoning process can worsen performance in others. We observe that the reason behind the performance drop-off is two-fold: (1) images generated from HTML are incorrect or missing information, and (2) generated images cannot depict the reasoning process.

As mentioned in the previous sections, VLM is not good at showing/tracking swaps, additions, or deletions in the HTML. Therefore, without images, responses are better when the questions have swaps, insertions, and deletions of elements (TRACKING SHUFFLED OBJECTS). In DATE UNDERSTANDING and NAVIGATE tasks, the images generated from HTML often fail to accurately represent the questions. In DATE UNDERSTANDING, LLAVA-1.5 generated HTML can not fully maintain the date, month, and year pattern mentioned in the question text ([Table 12.13](#)) which further confuses the VLM while performing reasoning with the image. Similarly, in NAVIGATE, GEMINI PRO generated HTML can not effectively depict the progression of navigation steps mentioned in the question text.

Image helps a different subset of a particular task. We further investigate the performance of with-image and without-image setups to find out when having an image is beneficial and when having an image hurts. As shown in [Figure 12.13](#), we break down the performance by tasks and count the number of times the VLM produces a correct answer with an image and gets the same question wrong without an image (*Image Improves*). Then, we reverse the conditions, i.e., count the number of times the VLM produces an incorrect answer with an image and generates the correct answer for the same question without an image (*Image Hurts*). We can see that for all tasks, having images helps solve some questions that can not be solved without images and vice versa.

4.5 Related Works

Visual Problem Solving. Visual problem-solving is an inherent human ability while performing complex multi-step reasoning. Humans tend to draw a mental image of a question to understand and ground the problem, which helps to plan subsequent steps to solve the question. Mental images provide a simplified representation of the content of the cognitive task. Thus, the involvement of visual images in problem-solving may result in a notable degree of success ([Bauer & Johnson-Laird, 1993](#); [Antonietti, 1991](#)). Studies also

show that an accurate visual representation of a problem enhances the chances of solving the problem rather than having no representation at all (van Garderen et al., 2018; Krawec, 2014).

Reasoning with LLMs and VLMS. In recent years, several LLMs and VLMS have been introduced, which are showing impressive performance in complex reasoning tasks (OpenAI, 2023; Touvron et al., 2023a; Chowdhery et al., 2022; Liu et al., 2023b; Zhu et al., 2023a; Li et al., 2023a; Dai et al., 2023b; Liu et al., 2023d). However, when it comes to solving math word problems (Cobbe et al., 2021a; Koncel-Kedziorski et al., 2016; Patel et al., 2021) or symbolic reasoning tasks (Suzgun et al., 2022), the VLM can not fairly compete with the LLM as the nature of these tasks is unimodal. While considerable efforts have been invested in improving the performance of LLMs on these reasoning tasks during inference (Madaan et al., 2023; Wang et al., 2023a; Gao et al., 2023; Wei et al., 2023; Poesia et al., 2023; Hao et al., 2023), fewer endeavors have been made to tackle these challenges from the perspective of a vision-language model (Lee et al., 2023; Hsu et al., 2023). A very relevant work to ours is Hsu et al. (2023), which leverages LLM to generate drawing commands and reads out abstractions from the resulting picture. However, it relies on a fine-tuned visual foundation model (Lee et al., 2023) to interpret abstractions from the drawn diagram, requiring additional training data. In addition, diagrams can only benefit specific tasks, limiting their applicability to diverse reasoning types. In this paper, we study these text-only benchmarks using VLMS by proposing a simple idea to leverage the full potential of a VLM on diverse reasoning tasks.

4.6 Summary

In this chapter, we present SELF-IMAGINE, an approach that maximizes the capabilities of Vision Language Models (VLMS) in solving text-only reasoning tasks. Our method draws on a common human problem-solving technique, creating visual representations of problems to aid in reasoning. Our approach is self-sufficient, requiring no additional data, supervision, or training. Through our intensive experiments with diverse reasoning tasks, we find that SELF-IMAGINE significantly improves the performance of state-of-the-art VLMS (LLAVA-1.5 & GEMINI PRO) using self-generated images. We also find that the extent of improvement relies heavily on the quality of the generated image. Analogous to how humans draw different imaginations for different question types, we present cases where image improves and hurts the performance of the VLM, motivating future research on better image generation approaches.

Chapter 5

Structured Data Synthesis for Efficient Pretraining

The utility of synthetic data to enhance pretraining data quality and hence to improve downstream task accuracy has been widely explored in recent large language models (LLMs). Yet, these approaches fall inadequate in complex, multi-hop and mathematical reasoning tasks as the synthetic data typically fails to add complementary knowledge to the existing raw corpus. In this chapter, we propose a novel large-scale and diverse **Math Informed syNthetic Dialogue (MIND)** generation method that improves the mathematical reasoning ability of LLMs. Specifically, using MIND, we generate synthetic conversations based on OpenWebMath (OWM), resulting in a new math corpus, MIND-OWM. Our experiments with different conversational settings reveal that incorporating knowledge gaps between dialog participants is essential for generating high-quality math data. We further identify an effective way to format and integrate synthetic and raw data during pretraining to maximize the gain in mathematical reasoning, emphasizing the need to restructure raw data rather than use it as-is.

5.1 Task Overview

The ability to reason is a fundamental element of human cognition, encompassing our ability to think logically, draw conclusions, and make decisions based on available information (Gendron et al., 2024). Large Language Models (LLMs) have demonstrated remarkable performance across wide range of general reasoning and specialized knowledge tasks. In particular, the improvement of LLMs in solving complex mathematical reasoning tasks (Hendrycks et al., 2021c; Cobbe et al., 2021a) has been significant in recent years (Gemini, 2024; Nvidia et al., 2024; OpenAI, 2024b).

Strong mathematical reasoning ability heavily relies on the abundance of high-quality, composite, and structured pretraining corpora. An effective mathematical corpus should not only contain relevant content but also be formatted to guide models break down complex problems into smaller sub-problems and solve each part step-by-step—enhancing the model’s ability to process and reason about complex problems (Wei et al., 2022). Prior studies show that structured and well-formatted corpora play a crucial role in enhancing multi-hop and logical reasoning abilities (Cobbe et al., 2021a; Li et al., 2023b; Gunasekar et al.,

2023), underscoring the importance of well-organized mathematical datasets in pretraining LLMs.

Curating complex, high-quality structured mathematical data is costly and resource-intensive, largely due to the uneven distribution of high-quality sources. Most advanced models (OpenAI, 2024b; Gemini, 2024) are not publicly accessible, and it is unclear how their approach is enhancing math reasoning. To mitigate this challenge, synthetic data generation has emerged as a scalable, and cost-effective alternative for creating a more balanced and diverse training corpus for pretraining LLMs (Maini et al., 2024a; Eldan & Li, 2023; Gunasekar et al., 2023; Shah et al., 2024). However, while these techniques have shown promise in improving general reasoning tasks, their data often lack the step-by-step problem solving structure crucial for multi-hop reasoning and complex mathematical tasks (Maini et al., 2024a), making them sub-optimal for such reasoning.

To address these challenges, we propose **MIND**, a novel approach to generate **Math Informed syNthetic Dialogue** data at scale. In MIND, we provide a pretrained LLM with a web document and explicitly prompt it in a zero-shot manner to generate a conversation that—(a) decomposes the original context step-by-step into multi-turn conversations and (b) explores each step in depth within a single turn. As illustrated in Figure 5.2, MIND generates conversation from a raw text by prompting an open-source LLM on seven diverse conversational styles. The generated conversations are refined using heuristic filters and then can be used to pre-train a language model.

MIND demonstrates that transforming raw web text into structured conversations using an off-the-shelf open-source LLM significantly enhances the mathematical and logical reasoning abilities of LLMs compared to unstructured raw or rephrased web text. Additionally, provides the flexibility to preserve the diversity of the web corpora and leverage knowledge imbalances between participants for further expansion of the corpora as they either educate each other or collaboratively bridge their shared knowledge gaps through explanation and analysis in a conversation. Moreover, MIND enables the continuous generation of synthetic data from a single document by employing infinite conversational styles, further enriching the diversity. Unlike static text rephrasing (Maini et al., 2024a), conversations encourage dynamic reasoning, where participants build on each other’s ideas, ask questions, and offer clarifications. This quality makes conversations particularly effective for complex reasoning tasks, as they not only preserve the original information but also expand it with new layers of understanding and explanation.

In summary, the key contributions of this work are as follows:

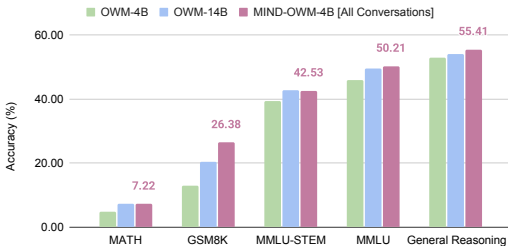


Figure 5.1: Continuous pretraining with all styles of conversations (MIND-OWM-4B) derived from a small subset (OWM-4B) and a $3.6\times$ large raw corpus (OWM-14B) reveals that model trained with conversations outperforms the one trained with larger corpus in GSM8K, MMLU and general reasoning—showing the significance of high-quality structured data over quantity.

- We propose a novel approach, MIND, to generate structured conversational synthetic data for math reasoning. Leveraging MIND, we produce 64B tokens of synthetic data using 14B tokens from OpenWebMath corpus.
- We conduct comprehensive experiments with various conversational styles, altering participant roles to assess their impact on conversation quality and reasoning tasks. Our findings emphasize the importance of the knowledge imbalance between participants in producing high-quality mathematical data.
- We scale our approach to higher number of tokens and to two math specific datasets, demonstrating its efficacy in large and high-quality raw corpus.
- We demonstrate an effective way for integrating synthetic and raw data during pretraining to enhance mathematical reasoning ability of LLMs, emphasizing the importance of carefully reformatting raw data to optimize reasoning processes instead of using it in its original form.

5.2 Methodology

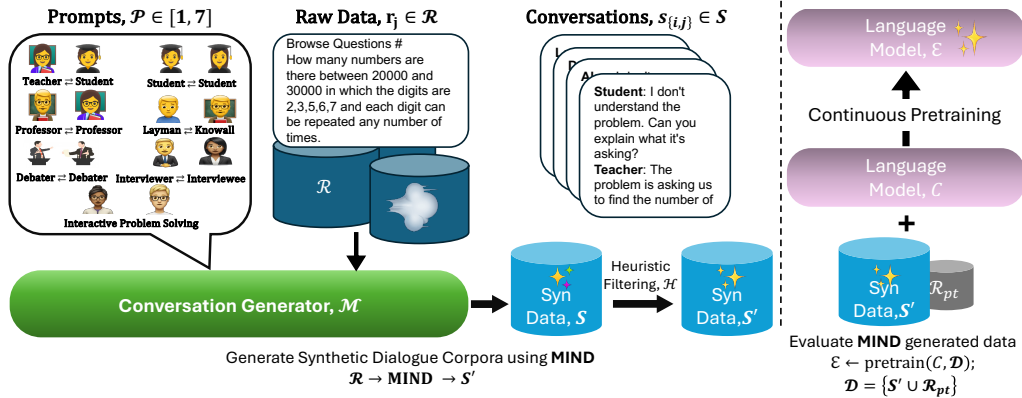


Figure 5.2: **Math Informed syNthetic Dialogue.** We (a) manually design prompts of various conversational styles, (b) provide the prompt along with raw context as input to LLM to obtain diverse synthetic conversations, (c) apply heuristic filtering to refine the generated data and (d) observe the downstream task performance after continuously pretraining an 7B LLM.

To generate high-quality data at scale, current synthetic data generation approach explores rephrasing texts using LLMs in varied syntax while preserving the core content (Maini et al., 2024a). However, their proposed approach limits up-sampling high-quality data in a way that does not go beyond grammatical styles or surface form transformations—leading little to no improvement when it comes to performance across complex and logical reasoning tasks. We hypothesize that simple rephrasing does not leverage the full potential of the synthetic data to improve the mathematical and complex multi-hop reasoning ability of LLM. Therefore, we propose, **MIND**, a conversational synthetic data generation approach that adds semantic variations and structured complexity to the raw text which is required to improve complex reasoning ability of the LLMs. In addition, multi-turn conversations can break down the original context step-by-step while each step addresses a sub-context at a time by often injecting complimentary reasoning or explanations. This resonates with how human solves a complex problem using consecutive chain-of-thought reasoning.

As depicted in [Figure 5.2](#), given a raw dataset $\mathcal{R} = \{r_1, \dots, r_N\}$, we define a set of conversational prompts $\mathcal{P} = \{p_1, \dots, p_7\}$ and utilize a pretrained LLM, denoted as \mathcal{M} , for synthetic data generation. We combine raw data r_j with a prompt p_i and pass it to \mathcal{M} to produce synthetic conversation $s_{i,j}$.

$$s_{i,j} = \mathcal{M}(p_i \parallel r_j)$$

Here, $s_{i,j}$ represents the synthetic data generated by applying prompt p_i to the raw example r_j . For a specific prompt, the total synthetic data generated can be represented as

$$\mathcal{S} = \{s_{i,j} \mid j \in [1, N]\} \text{ for a fixed } i \in [1, 7]$$

We further apply heuristic filtering (\mathcal{H}) to remove bad generations:

$$\mathcal{S}' = \mathcal{H}(\mathcal{S})$$

Finally, we have a high-quality synthetic dialogue corpus \mathcal{S}' which is specifically designed to improve mathematical and logical reasoning ability. To summarize MIND:

$$\mathcal{R} \rightarrow \text{MIND} \rightarrow \mathcal{S}'$$

To evaluate the effectiveness of \mathcal{S}' in pretraining, we conduct continuous pretraining on a base LLM, \mathcal{C} , to minimize the computational costs associated with full pretraining. Our prior experiments on complete pretraining with raw data, \mathcal{R} and synthetic data, \mathcal{S}' validates that the ranking between models trained on \mathcal{S}' or \mathcal{R} remains consistent whether we use continuous pretraining or full-scale pretraining (detailed in [Appendix 12.9.1](#)). Moreover, continuous pretraining has emerged as an effective way to improve performance of LLMs in target domains ([Guo et al., 2024b](#); [Huang et al., 2023](#); [Chen et al., 2023d](#)) and even boost their general capabilities ([Ibrahim et al., 2024](#); [Parmar et al., 2024c](#)) with reduced training cost. Given the similar outcomes and significant savings in computational resources, we adopt continued pretraining for evaluating our approach throughout the paper.

Using \mathcal{S}' and a subset of pretraining data (\mathcal{R}_{pt}), the model \mathcal{C} is continuously pretrained, yielding an enhanced model \mathcal{E} , which possess improved mathematical reasoning capabilities.

$$\mathcal{E} \leftarrow \text{pretrain}(\mathcal{C}, \mathcal{D}); \text{ where } \mathcal{D} = \{\mathcal{S}' \cup \mathcal{R}_{pt}\}$$

5.2.1 Compose Conversational Prompts

To generate conversation using a document r_i , we prompt \mathcal{M} in a way that preserves all information from the original context in the conversation and remains faithful to the context. We manually compose several prompts on diverse conversation setting and topics. We finalize seven prompts (\mathcal{P}) featuring conversations between (1) TWO STUDENTS, (2) TEACHER STUDENT, (3) TWO PROFESSORS, (4) DEBATE, (5) PROBLEM SOLVING, (6) LAYMAN KNOWALL, and (7) INTERVIEW which can be found in [Appendix 12.8.1](#). These prompts are specifically designed to guide LLM in breaking down the input context step-by-step, with each step being discussed in depth through explanations and reasoning.

In this work, we mostly consider only two-party, sequential conversations over a single document or problem. We do *not* study more complex decision processes such as **hierarchical decision-making by a group of experts**, **multi-party** (more than two) interactions, or **partly parallel** deliberation (e.g., sub-teams reasoning in parallel and then merging). The same recipe—prompts that assign roles and instruct \mathcal{M} to produce a faithful, step-wise

discussion—could in principle be extended to such settings (e.g., the DEBATE prompt can in practice yield multiple debaters in a single thread); that would require new prompt designs and we leave it to future work.

5.2.2 Generate Conversation

Given an unstructured raw text (r_j), we instruct the LLM to convert the raw text into a multi-turn conversation ($s_{i,j}$) using a prompt (p_i) where $p_i \in \{\text{two_students}, \text{teacher_student}, \dots, \text{debate}\}$.

Seed Data Selection. The benefit of MIND will get amplified for raw texts that require step-by-step analysis and chain of thought reasoning—the key features of a math corpus. Therefore, we choose OpenWebMath (Paster et al., 2023) as our seed corpus, \mathcal{R} , which contains 14.7B tokens of high quality mathematical web text.

Large Language Model. We use $\mathcal{M} = \text{LLAMA3-70B-INSTRUCT}$ (AI@Meta, 2024) to generate conversations from raw text, due to its superior performance across a variety of tasks compared to other open-source models. The instruction-tuned version is specifically fine-tuned and optimized for dialogue and chat-based applications.

Generation Configuration. We observe that with increasing context length, conversations tend to lose details from the original texts, as discussed in Appendix 12.10.1. Therefore, for each generation, we iteratively take contexts of 500 tokens to obtain accurate and informative conversations. To evaluate the quality of the generated conversations, we test various filtering methods, from simple heuristics to LLM-based scoring. However, as noted in Appendix 12.10.3, LLM scoring consistently rates all generations highly, making it unsuitable for our approach. Hence, we rely on heuristic filtering to discard bad generations before using them for training.

5.3 Experimental Setup

Conversation Generator Configuration. To generate conversation, we consider zero-shot prompting \mathcal{M} , where we only pass a basic prompt (Appendix 12.8.1) and the raw text. We sample conversations with temperature=1.0 and top_p=0.9 where the total number of input-output tokens is limited to 4096. We use the TensorRT-LLM toolkit to deploy large scale generation¹.

Pretrained Model Architecture. We train a standard decoder-only Transformer (Vaswani et al., 2017a) architecture of 7B parameters (C). The framework uses causal attention masks and Rotary Position Embeddings (RoPE) (Su et al., 2021), Tiktoken tokenizer, SwiGLU (Shazeer, 2020) activations in the MLP layers, and grouped query attention (GQA) (Ainslie et al., 2023). The model consists of 32 layers, 32 attention heads, sequence length of 4096, and a hidden dimension size of 4096. It has no bias terms, has dropout rate of zero, and untied input-output embeddings. The models are trained using NVIDIA’s Megatron-LM (Shoeybi et al., 2019b) repository.

¹<https://github.com/NVIDIA/TensorRT-LLM>

5.3.1 Training Details

Pretraining Data. Our pretraining data blend comprises of publicly available datasets from 13 snapshots of CommonCrawl (73.37%) (Gao et al., 2020b), books/patents (9%), papers (9%), code (5.12%), stack-exchange (2.66%), and Wikipedia (0.8%). Our code data consists of 42 programming languages while the other datasets come from various sources including web documents, news articles, scientific papers, and books.

General Pretraining. To prepare a base model, we pretrain a 7B LLM on our pretraining data blend till 700B tokens using 512 H100 80GB SXM5 GPUs. During training, we use the AdamW optimizer (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and weight decay of 0.1. We use a 2-way tensor and pipeline parallelism to train the model. We set the maximum value of learning rate to $3e^{-4}$, minimum to $3e^{-6}$, and use a batch size of 6M tokens with a 4096 context length.

Continued Pretraining. After pretraining the base model (\mathcal{C}) on 700B tokens, we proceed with continuous pretraining using an additional 50B tokens to obtain \mathcal{E} . To reduce the shift between pretraining and continuous pretraining token distributions (Guo et al., 2024b) we create a new data blend (\mathcal{D}) for this phase. To ensure the model is exposed to more math tokens, blend \mathcal{D} consists of 2:1 ratio of OpenWebMath (33B tokens)—either raw (\mathcal{R}) or synthetic (\mathcal{S}')—and 13 snapshots of CommonCrawl (17B tokens) (\mathcal{R}_{pt}) to maintain consistency with the pretraining blend. To ensure fair comparison, we always keep this token distribution constant in every experiment i.e., every model will see a the same amount of tokens from a data source regardless of its size. Unlike the pretraining blend, we use a high quality version of CommonCrawl data (\mathcal{R}_{pt}) filtered by the FineWebEdu (Penedo et al., 2024) classifier to achieve reasonable performance in generative tasks. This \mathcal{R}_{pt} remains constant across all our continued pretraining experiments, while we vary the OpenWebMath with \mathcal{R} or \mathcal{S}' or combining both to assess their relative significance. We maintain the same training configuration as before and continue pretraining until reaching 50B tokens, using the same pretraining loss objective. In this paper, we use two versions of OpenWebMath:

- **OWM-4B:** To quickly evaluate the effectiveness of all seven prompts, we take a smaller subset of OpenWebMath containing 4B tokens. Synthetic data generated from this subset is labeled as -4B throughout the paper.
- **OWM-14B:** This version contains the full 14.7B tokens of OpenWebMath and the synthetic data of this is called -14B.

5.3.2 Evaluation Metrics

To evaluate the zero-shot and few-shot learning capabilities of our models, we conduct a thorough benchmark assessment using a series of datasets using LM Eval Harness (Gao et al., 2024b).

General Purpose Reasoning Tasks. This category comprises datasets testing broader cognitive skills and language comprehension. We consider nine standard commonsense and logical reasoning tasks in 0-shot: ARC easy (ARC-E) & challenge (ARC-C) (Clark et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), OpenBookQA (Mihaylov et al., 2018), TruthfulQA (Lin et al., 2022), CommonsenseQA (Talmor et al., 2019) and a reading comprehension task: RACE (Lai et al.,

2017). We report the average results across ten general reasoning tasks under the metric ‘GENERAL REASONING’.

Math and Specialized Knowledge Tasks. We consider three diverse math benchmarks to comprehensively evaluate the mathematical reasoning ability. These benchmarks encompass mathematical challenges from elementary to college level complexity demanding qualitative reasoning (8-shot GSM8K (Cobbe et al., 2021c), 4-shot MATH (Hendrycks et al., 2021e)) and conceptual science and math reasoning (5-shot MMLU-STEM (Hendrycks et al., 2021b)). In the Specialized Knowledge category, we evaluate on MMLU that spans multiple domains, from professional to academic, testing the model on specialized subjects.

5.4 Experiments and Results

By leveraging MIND with seven conversational prompts and the raw OWM-4B, we generate a new corpus of 43 billion tokens (All Conversations). Additionally, employing the entire OWM-14B dataset and TWO STUDENTS conversation style, MIND produces an additional 21 billion tokens—resulting in a total of 64 billion tokens. This underscores MIND’s potential to generate vast amount of high-quality data from relatively limited source material².

Performance across Individual Prompt Style. We observe the effect of each conversation style by generating synthetic data with seven prompts for a smaller subset of Open-WebMath, denoted as OWM-4B. To establish a baseline, we continue pretraining \mathcal{C} using $\mathcal{D} = \{\mathcal{R} \cup \mathcal{R}_{pt}\}$, where $\mathcal{R} \in$ OWM-4B. To further assess the significance of MIND over other synthetic data generation approach, we add another baseline ‘Rephrase’ introduced by Maini et al. (2024a). We generate rephrases with \mathcal{M} using the highest performing prompt from their paper to maintain consistency among generation quality and training setup. We continuously train \mathcal{C} with \mathcal{D} where $\mathcal{R} \in$ Rephrase-OWM-4B. In subsequent experiments, we replace \mathcal{R} with \mathcal{S}' where $\mathcal{S}' =$ MIND-OWM-4B, corresponding to a particular conversation style, and repeat the training. To assess the utility of combining multiple conversations, we create a new dataset by selecting the longest conversation for each context from the seven generated conversations, labeling it as the LONGEST CONVERSATION dataset.

As shown in Table 5.1, models trained on MIND-generated data of individual styles consistently outperform those trained on rephrased or raw data across all reasoning tasks. Specifically, models trained on synthetic data exhibit significant improvements in mathematical reasoning compared to the baseline, achieving absolute gains ranging from 4.78% to 12.82% on GSM8K, 0.54% to 1.28% on MATH, and 0.79% to 4.28% on MMLU-STEM. In specialized knowledge tasks such as MMLU, synthetic data leads to improvements ranging from 1.08% to 4.55%. Furthermore, synthetic data yields an overall enhancement in general reasoning ability, with up to a 2% absolute average improvement across the ten reasoning tasks. The LONGEST CONVERSATION delivers the highest gains across all tasks, demonstrating the potential of incorporating multiple perspectives into the training corpus.

The disparity between Rephrase and MIND is closely related to the limitations of the rephrasing process. Rephrase adds linguistic variations to the older data, preserving the syntactic meaning of the document, but can not generate semantic/pragmatic variations. Moreover, rephrases are limited to the information in the raw text and unable to inject new

²To maintain consistency, we use a subset of the data (33B tokens) in all experiments.

³Further breakdown of individual tasks are in Appendix 12.9.2.

Dataset	Style	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-All*
OWM-4B	Raw	12.96	4.92	39.39	45.91	52.90	29.17
	Rephrase	11.68	5.46	39.71	46.17	53.58	29.22
MIND-OWM-4B	TEACHER STUDENT	22.74	5.96	40.72	47.93	54.84	32.87
	TWO STUDENTS	21.30	6.20	41.90	48.77	54.32	32.65
	LAYMAN KNOWALL	17.74	5.46	41.96	48.87	54.89	31.74
	DEBATE	23.96	6.12	40.18	47.61	54.76	33.11
	INTERVIEW	20.92	5.86	40.53	46.99	54.73	32.12
	PROBLEM SOLVING	24.72	6.16	41.36	47.74	54.90	33.38
	LONGEST CONVERSATION	25.78	6.30	42.72	49.37	54.86	34.08

Table 5.1: **Results of 7B LLM pretrained on Diverse Conversational Styles.** Continuous training with different conversation styles improves all reasoning tasks. Selecting the longest conversation for each raw text further enhances performance in math and specialized knowledge tasks³. *Average of GSM8K, MATH, MMLU and General Reasoning.

knowledge into the data. As evidenced in our experiments, while rephrasing offers some benefits, it falls short in addressing the deeper, more complex reasoning challenges that conversational data can resolve. The structured and interactive nature of conversations facilitates a more nuanced understanding of the problem space, making it an effective approach for improving mathematical reasoning of LLMs.

Analysis with Complete OpenWebMath. Building on the findings from OWM-4B experiments, we establish that all seven conversational styles contribute to significant improvements compared to the raw data. This insight prompted us to explore the effect of increased data in reasoning by scaling our synthetic conversation generation for the complete OWM-14B corpus. To generate data, we follow the similar recipe as before and apply only one conversation style to minimize the generation cost. Among the top three highest-performing prompts across all tasks, we randomly choose TWO STUDENTS prompt style to generate conversations (MIND-OWM-14B). We then continuously train \mathcal{C} on OWM-14B and MIND-OWM-14B alternatively to assess the impact at a larger data scale. In this phase, we include another experiment by continuously training \mathcal{C} on 50B additional tokens using $\mathcal{D} = \{\mathcal{R}_{pt}\}$ to observe how much gain we can attain across all tasks from math-centric pretraining.

Dataset	Style	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-All
Pretraining Data	Raw	9.33	4.74	37.84	45.41	53.22	28.17
OWM-14B		20.47	7.24	42.82	49.49	53.95	32.79
MIND-OWM-14B	TWO STUDENTS	27.29	8.24	43.55	49.91	55.54	35.25

Table 5.2: **Results of 7B LLM trained on Complete OWM-14B and MIND-OWM-14B:** Continuous training of LLM with synthetic conversation outperforms models trained with original pretraining blend and raw OpenWebMath across all tasks.

As consistent with the previous findings, Table 5.2 shows that model trained on synthetic conversations is undoubtedly the best for math benchmarks while it also improves overall average for all other reasoning tasks. This underscores that, with data scaling, MIND maintains significant gains in mathematical reasoning while preserving and enhancing

performance across other reasoning tasks, including commonsense, factual, and specialized knowledge.

5.5 Ablations

Does the Prompt Style matter? From Table 5.1, we observe improvement across all tasks using six conversational styles. However, our experiment with TWO PROFESSORS conversations yield relatively equivalent or worse performance compared to the raw data (Table 5.3).

Dataset	Style	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-All
OWM-4B	Raw	12.96	4.92	39.39	45.91	52.90	29.17
MIND-OWM-4B	TWO PROFESSORS	13.50	4.52	37.93	45.25	53.21	29.12

Table 5.3: **TWO PROFESSORS prompt style vs Raw data.** Continuous pretraining with TWO PROFESSORS conversations does not provide gain over raw data compared to other conversational styles.

This outcome can be attributed to the nature of the TWO PROFESSORS conversation style. Upon reviewing the generated conversations, we hypothesize that the relatively lower performance is due to the zero-knowledge gap between participants. In this setup, both participants assume that the other already has sufficient knowledge as they are the domain experts, leading to surface-level engagement and less detailed discussions.

To further investigate, we measure the BLEU and ROUGE scores between the raw text and the corresponding conversation, as shown in Figure 5.3, and find that the TWO PROFESSORS style exhibits the highest similarity to raw text. This implies that TWO PROFESSORS dialogues do not fully exploit the potential of the generation model to introduce new reasoning or breakdowns of complex problems, aligning with our qualitative observation that the professors are not engaging in deeper analysis of concepts. This contrasts with other conversational styles where there is either a clear knowledge gap between participants (LAYMAN KNOWALL, TEACHER STUDENT, INTERVIEW), forcing one to explain concepts in more depth, or both participants, being non-experts are actively analyzing and solving the problem (PROBLEM SOLVING, DEBATE, TWO STUDENTS) which results in expanded dialogues with complementary explanations and reasoning. In the latter case, the lack of expertise creates an implicit knowledge gap—instead of one participant being more knowledgeable, both non-experts collaborate to bridge their shared knowledge gap. As depicted in Figure 5.3, the LAYMAN KNOWALL style, which features the greatest knowledge imbalance between participants, has the lowest BLEU and ROUGE scores. This supports our hypothesis that

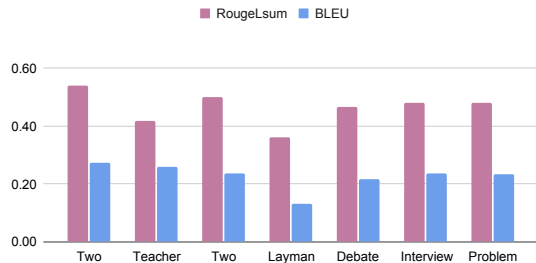


Figure 5.3: **Similarity between Raw Text & Synthetic Dialogues.** The TWO PROFESSORS style exhibits greater similarity to raw text, while LAYMAN KNOWALL shows the lowest similarity due to its richer context with details and explanations.

a larger information gap encourages the knowledgeable participant to explain concepts thoroughly, leading to more explicit and detailed conversations.

Relating these insights to our findings in Table 5.1, we see that incorporating explicit knowledge gaps in dialogues is beneficial for MMLU and general reasoning tasks. Conversely, collaborative problem solving, to close the implicit knowledge gap, is crucial for improving performance on math tasks. This highlights a key characteristic of high-quality math data—merely breaking down the problem is insufficient for effective math reasoning. Instead, dynamic knowledge exchange and analysis within the dialogues are essential to achieve maximum improvement in math reasoning.

Does Conversation benefit other datasets? OpenWebMath used in our current experiments is predominantly collected from mathematical web pages that can contain noisy web contexts. Generating synthetic conversations for such noisy contexts upsamples high-quality data and hence we observe a huge gain in performance with high-quality conversations. Here, we investigate if MIND works on high-quality datasets such as books or papers. We consider a new seed corpus, MATHPILE (Wang et al., 2023b), that consists of 9.3B tokens extracted from high-quality data sources such as ArXiv papers, textbooks, StackExchange, Wikipedia, ProofWiki, and CommonCrawl pages.

Dataset	Style	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-All
Pretraining Data	Raw	9.33	4.74	37.84	45.41	53.22	28.17
MATHPILE		8.79	4.96	42.82	49.49	54.16	29.35
MIND-MATHPILE	TWO STUDENTS	12.74	5.74	43.55	49.91	53.98	30.59

Table 5.4: **MATHPILE vs Synthetic Conversation from MATHPILE (MIND-MATHPILE).** Conversation generated from high-quality raw data further improves the performance of math tasks.

By employing \mathcal{M} , we generate conversations from raw text with the TWO STUDENTS prompt. Later, we replicate the experiments by replacing OWM with MATHPILE and MIND-MATHPILE accordingly. Table 5.4 shows that MIND-MATHPILE outperforms the raw counterpart in all three math benchmarks along with specialized knowledge tasks, achieving comparable scores in general reasoning task. In addition, majority of MATHPILE data is from ArXiv papers and recent work has found this source ineffective in improving mathematical reasoning (Zhihong Shao, 2024). We observe a similar trend, where non-math focused pretraining corpora yields better GSM8K score than raw MATHPILE corpus. However, our synthetic conversation on MATHPILE rather amplifies the quality of the corpus resulting in 3.95% absolute improvement on GSM8K in comparison with raw data. This highlights the superior structured complexity of conversations, which proves particularly effective for multi-hop and mathematical reasoning tasks, over high-quality data from ArXiv papers.

The difference in gains between MATH and GSM8K raises the question of why MIND MATHPILE is superior in one over the other. It is worth noting that 85% of MATHPILE constitutes of scientific ArXiv papers. Even in (Wang et al., 2023b), they observe that MATH gain is minimal or worse with MATHPILE data. We attribute this performance degradation to the disparity between the math knowledge present in ArXiv papers and that required for the downstream benchmarks. GSM8K problems are elementary-to-middle school level. What separates good

from bad performance is whether the model learned to reason step-by-step, explain its thinking, and structure a solution narrative. Converting raw ArXiv papers into conversations likely injects exactly this: dialogue turns that model how to walk through a problem, ask clarifying questions, and articulate reasoning chains. MATH contains competition-level problems that require genuine advanced mathematical knowledge—like number theory tricks, combinatorial identities, or non-obvious algebraic manipulations. The ArXiv papers in MATHPILE, even when reformatted conversationally, is largely research-level mathematics that doesn’t directly map to competition problem-solving skills. Conversational scaffolding can fix how a model reasons; it can’t easily inject what mathematical tools it knows. In short: MIND helps most when the bottleneck is reasoning format, and helps less when the bottleneck is domain knowledge depth.

Is replacing with Synthetic Data the best option? Our findings in Table 5.1, 5.2 indicate that completely replacing OpenWebMath with synthetic data yields the best performance across benchmarks. However, Maini et al. (2024a) emphasizes the importance of combining real data and synthetic rephrases to achieve consistent improvements across a broader range of tasks—a similar trend we observe in our experiment with rephrased data, as shown in Table 5.5. To investigate this further, we conduct experiments with four data combinations using OWM-4B while the \mathcal{R}_{pt} remains constant:

- **OWM-4B + MIND-OWM-4B [1:1].** We combine \mathcal{R} and \mathcal{S}' in a 1:1 ratio, ensuring an equal number of tokens to be seen during pretraining from both sources. For the synthetic data, we utilize the LONGEST CONVERSATION, as this shows the most improvement across tasks (Table 5.1).
- **OWM-4B + MIND-OWM-4B [Concat].** We concatenate each raw context with all seven synthetic conversations sequentially.
- **MIND-OWM-4B [Longest Conversation].** From the seven conversations generated for each context, we select the longest conversation in token count.
- **MIND-OWM-4B [All Conversations].** This data incorporates all conversation across all styles.

Dataset	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING	
					(Avg)	Avg-All
OWM-4B	12.96	4.92	39.39	45.91	52.90	29.17
OWM-14B	20.47	7.24	42.82	49.49	53.95	32.79
Rephrase-OWM-4B	11.68	5.46	39.71	46.17	53.58	29.22
OWM-4B+Rephrase-OWM-4B [1:1]	14.25	6.20	42.31	48.74	53.68	30.72
OWM-4B+MIND-OWM-4B [1:1]	21.68	6.14	42.56	49.57	54.50	32.97
OWM-4B+MIND-OWM-4B [Concat]	24.49	6.22	43.67	50.46	55.10	34.07
MIND-OWM-4B [Longest Conversation]	25.78	6.30	42.72	49.37	54.86	34.08
MIND-OWM-4B [All Conversations]	26.38	7.22	42.53	50.21	55.41	34.80

Table 5.5: **Comparison of 7B LLM trained with raw and combination of synthetic data.** Synthetic conversation outperforms raw data in all combinations. Specifically, combinations of all conversations generated from OWM-4B surpasses the performance of OWM-14B ($3.6\times$ larger corpus) across all tasks, underscoring the superior quality and diversity of the conversations.

Our finding in [Table 5.5](#) indicates that all combinations provide substantial boost in performance across all tasks. However, for math-centric benchmarks (GSM8K and MATH), training solely with synthetic conversations elicits the best improvements. This is likely as these tasks require complex and multi-step reasoning and conversations are designed to replicate these type of thinking. In parallel, having both raw data and conversation is beneficial for specialized and general purpose reasoning tasks, aligning with the findings in [Maini et al. \(2024a\)](#). Since synthetic data tends to remove special tags, styles, and code indentations, the inclusion of raw data helps improve the generalizability of LLMs across diverse domains. Additionally, to measure the maximum gain we can achieve from conversations for a limited data, we continuously train \mathcal{C} with all synthetic dialogues generated from OWM-4B. As shown in [Table 5.5](#), using conversations generated from OWM-4B, we can outperform the model trained with $3.6\times$ bigger corpus (OWM-14B) on GSM8K, MMLU and general reasoning tasks while showing comparable performance on other tasks. Inspired by this, we further compare MIND with DEEPSEEK MATH ([Zhihong Shao, 2024](#)) that extract 120B unique math tokens from CommonCrawl ([Appendix 12.10.4](#)). The results from [Table 12.21](#) demonstrate that diverse conversations from MIND based on a small seed corpus can yield comparable math accuracy to the DEEPSEEK MATH model. This illustrates the potential to enhance reasoning with limited data by generating synthetic conversations of infinite styles.

Does the improvement persist with smaller \mathcal{M} ? In the previous experiments, we used a constant \mathcal{M} , a powerful instruction-tuned model. However, it remains unclear whether the improvements in downstream reasoning tasks stem from the quality of the generated dialogues or are primarily due to model distillation from the powerful LLM. To assess the impact of \mathcal{M} on the downstream task performance, we re-run MIND with a smaller \mathcal{M} =LLAMA3-8B-INSTRUCT on PROBLEM SOLVING style, the best performing style in [Table 5.1](#) and continuously pretrained a 7B LLM following the training setup in [Section 5.3.1](#).

Dataset	\mathcal{M}	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-All
OWM-4B	-	12.96	4.92	39.39	45.91	52.90	29.17
MIND-OWM-4B	LLAMA3-8B-INSTRUCT	22.37	5.72	41.36	48.48	55.21	32.95
	LLAMA3-70B-INSTRUCT	24.72	6.16	41.36	47.74	54.90	33.38

Table 5.6: Results of 7B LLM trained on MIND-OWM-4B using \mathcal{M} of different sizes: Regardless of the sizes of \mathcal{M} , model trained on MIND-OWM-4B outperforms the one trained with raw data.

As shown in [Table 5.6](#), even with a smaller \mathcal{M} , the MIND-generated data provides a significant boost in math and general reasoning abilities compared to the raw/rephrased data. This demonstrates that the gains are not solely dependent on the capabilities of the larger \mathcal{M} but are largely driven by the quality and structure of the MIND-generated dialogues. Additionally, regardless of model size and method of synthetic data generation, all LLM-generated synthetic data involves some form of knowledge distillation. However, we demonstrate an effective distillation approach that significantly enhances the reasoning ability of LLMs compared to existing approaches ([Maini et al., 2024a](#)).

5.6 Discussion: Why Role-Based Conversations Work?

The empirical variance across conversational roles in MIND is not merely a stylistic artifact—it reflects a deeper structural principle governing what makes synthetic data useful for pretraining. We argue that the effectiveness of a conversational style can be understood through a single unifying lens: *the degree to which a role assignment structurally forces the generation of reasoning that is complementary to the source text.*

Formalizing Complementary Information. Let r_j denote a raw document and $s_{i,j} = \mathcal{M}(p_i || r_j)$ the synthetic conversation generated from prompt p_i . Define the *complementary information* of a conversation as the content present in $s_{i,j}$ that is not recoverable from r_j alone—reasoning steps, explanations, analogies, or decompositions that the source document leaves implicit. We operationalize a proxy for this quantity using surface-level divergence:

$$\Delta_i = 1 - \text{sim}(s_{i,j}, r_j) \tag{5.1}$$

where $\text{sim}(\cdot, \cdot)$ is a lexical similarity measure such as BLEU or ROUGE-L. As shown in [Figure 5.3](#), Δ_i varies substantially across styles: TWO PROFESSORS yields the lowest divergence (highest BLEU/ROUGE), while LAYMAN KNOWALL yields the highest. Critically, downstream accuracy tracks Δ_i in a consistent direction—styles with higher complementary divergence tend to produce stronger reasoning gains. However, Δ_i alone is insufficient as an explanation, since LAYMAN KNOWALL has the highest Δ_i yet does not yield the best mathematical reasoning results ([Table 5.1](#)). This suggests that complementary information is a necessary but not sufficient condition: the *type* of complementarity matters as much as its quantity.

Two Regimes of Knowledge Gap. We propose that the seven conversational styles partition naturally into two functional regimes, each generating a distinct type of complementary content:

- **Explicit knowledge gap** (TEACHER STUDENT, LAYMAN KNOWALL, INTERVIEW): One participant possesses substantially more domain knowledge than the other. The structural asymmetry *obligates* the knowledgeable participant to externalize tacit reasoning—to explain concepts from first principles, scaffold understanding, and justify each step. The resulting data is rich in instructional reasoning traces.
- **Implicit knowledge gap** (TWO STUDENTS, PROBLEM SOLVING, DEBATE): Neither participant has full mastery, so both must reason *collaboratively and incrementally* to make progress. This creates a different kind of complementary content—exploratory, iterative, and analytically dense—that mirrors the problem-solving process itself rather than the transmission of established knowledge.

The failure case is illuminating: TWO PROFESSORS belongs to neither regime. When both participants are assumed to be domain experts, there is no communicative pressure to unpack or explain—agreement is reached quickly and at the surface level. This explains why the TWO PROFESSORS style exhibits BLEU and ROUGE scores closest to the raw text and yields no meaningful gains over the raw corpus baseline ([Table 5.3](#)).

Task-Type Alignment. The two regimes differ not only in the structure of the reasoning they generate but also in the types of downstream tasks they benefit. Let $\mathcal{T}_{\text{proc}}$ denote tasks that reward procedural reasoning (e.g., GSM8K, MATH) and $\mathcal{T}_{\text{know}}$ denote tasks that rely more heavily on conceptual or factual knowledge (e.g., MMLU). Empirically, we observe that collaborative styles with an implicit knowledge gap tend to produce stronger gains on procedural reasoning tasks, while explicit-gap styles often benefit knowledge-oriented benchmarks:

$$\text{Gain}(\mathcal{T}_{\text{proc}}, \text{implicit gap}) > \text{Gain}(\mathcal{T}_{\text{proc}}, \text{explicit gap}). \quad (5.2)$$

This pattern is consistent with the ablations reported in [section 10.5](#). Collaborative problem-solving styles (TWO STUDENTS, PROBLEM SOLVING, DEBATE) yield larger improvements on GSM8K, where success depends on constructing step-by-step solution traces. Explicit-gap styles such as LAYMAN KNOWALL, in contrast, generate more explanatory and expository content that enriches conceptual representations, which can be beneficial for broader knowledge and reasoning benchmarks such as MMLU.

Decomposition Diversity and Aggregation. Mathematical reasoning admits multiple valid decompositions—the same content can be explained linearly (TEACHER STUDENT), via alternatives and counterarguments (DEBATE), or through collaborative discovery (TWO STUDENTS). Each role p_i favors a different decomposition strategy; hence the set $\{s_{1,j}, \dots, s_{K,j}\}$ for fixed r_j spans a diverse subset of valid reasoning trajectories. Training on the union of role-induced conversations is analogous to data augmentation in the space of decompositions, improving coverage and reducing overfitting to a single exposition style. This aligns with the empirical gains from LONGEST CONVERSATION and ALL CONVERSATIONS in [Table 5.1](#) and [Table 5.5](#): ALL CONVERSATIONS outperforms OWM-14B with a $3.6\times$ smaller raw corpus, indicating that role-based variance increases effective data diversity without requiring more raw documents.

The Role of Structural Pressure. The unifying principle across all effective styles is that the role assignment must create *structural pressure*—a communicative necessity that cannot be satisfied without generating explicit reasoning. When one participant lacks knowledge, the other must explain; when both lack knowledge, they must reason together. In either case, the generation model is forced to produce content that goes beyond the surface of the source document. This structural pressure is what rephrasing fundamentally lacks: a rephrase preserves syntactic form while leaving the implicit reasoning structure of the original text untouched, which is precisely why rephrasing provides minimal gains on complex reasoning tasks despite using the same generation model. This analysis suggests a principled criterion for designing future conversational styles: a role assignment is likely to be effective for mathematical reasoning if and only if it structurally prevents participants from reaching agreement without first externalizing intermediate reasoning steps.

5.7 Related Works

Mathematical Data Curation. Selecting high quality data for pretraining LLMs is essential for producing state-of-the-art large language models (Brown et al., 2020a; Chowdhery et al., 2023; Parmar et al., 2024a;b; Rae et al., 2021; Feng et al., 2024a). Several mathematical datasets have been introduced in recent years (Paster et al., 2023; Wang et al., 2023b; Azerbayev

et al., 2023a; Welleck et al., 2021) which have been carefully collected from the web using different heuristics. OpenWebMath contains 14.7B tokens of mathematical web pages filtered from CommonCrawl based on math strings, \LaTeX contents and a math document classifier. Building on this corpus, DEEPSEEK MATH (Zhihong Shao, 2024) trains a fastText (Joulin, 2016) classifier to further extract mathematical documents from CommonCrawl. They cluster the extracted documents based on the URL domain and label a domain math-related where over 10% of the web pages have been collected are classified as math content. Finally, web pages linked to these URLs, yet uncollected, will be added to the seed corpus which will be used to retrain the fastText classifier to fetch diverse math contexts. MATHPILE (Wang et al., 2023b), a multi-source corpus (8.9B tokens), has been aggregated from textbooks, Wikipedia, ProofWiki, CommonCrawl, StackExchange, and arXiv, with the majority (over 85%) sourced from high quality data source arXiv. Although these datasets can effectively capture the diverse mathematical information from web, it is difficult to detect and filter out noisy dataset. Recently, many powerful models (OpenAI, 2024b; Jiang et al., 2023; Gemini, 2024; Anthropic, 2024; Team, 2024a), in addition to not open sourcing their data, are also refraining from disclosing detailed information about their corpus. For the open-source community, constructing high-quality and diverse pretraining corpora is a crucial factor in bridging the performance gap with closed-source models which is the main objective of our work.

Synthetic Math Data. Generating synthetic math data using LLMs has been widely explored in recent days (Trinh et al., 2024; Li et al., 2024a; Gunasekar et al., 2023; Madaan et al., 2024; Patel et al., 2024; Toshniwal et al., 2024b) specifically during alignment using supervised fine-tuning (SFT) (Taori et al., 2023). Some of the latest approaches focus on generating data from seed problems. For instance, Yu et al. (2023) rewrites existing benchmark questions from multiple perspectives using LLMs to create new mathematical problems, while Huang et al. (2024b); Shah et al. (2024) leverage GPT-4 to extract topics and key points from seed samples and recombine them into new questions. To further improve diversity, Chan et al. (2024) uses GPT-4 to generate questions and answers at scale, incorporating over one million personas. Previous approaches to generate synthetic data is primarily designed for fine-tuning rather than pretraining, distinguishing it from our effort. Similar to ours, Dai et al. (2022) converts documents into dialogues by predicting unobserved questions without altering the original document. However, MIND expands knowledge by adding complementary reasoning and explanations, leveraging diverse conversational styles to enhance reasoning and enrich diversity, which is infeasible with Dai et al. (2022). In the context of pretraining, recent works have generated synthetic datasets (Gunasekar et al., 2023; Li et al., 2023b) to train smaller language models that demonstrate equivalent performance as the larger models on certain mathematical benchmarks. However, these methods remain largely opaque, costly, and reliant on proprietary models to produce billions of tokens. Additionally, such data generation can be biased towards specifically generating data related to tasks that we want to perform well on. In contrast, MIND provides a feasible alternative to upsample high quality structured data from diverse web contexts, that embeds multi-step and chain-of-thought reasoning, using an off-the-shelf open source LLM.

5.8 Summary

In this chapter, we focus on improving the mathematical reasoning abilities of open-source LLMs through sustainable synthetic data generation approach. We propose a simple approach to generate complex and structured data at scale, called MIND, that produces a new conversational synthetic math corpus, MIND-OWM, using an off-the-shelf open-source LLM. Models trained on MIND-OWM, a corpus generated through our approach, consistently outperform those trained on raw data, achieving up to a 6.29% improvement across mathematical reasoning benchmarks and outperforming models trained on $3.6\times$ larger datasets. Importantly, these gains persist across general-purpose reasoning tasks and when scaling up the data, highlighting the versatility of synthetic conversations. This work demonstrates the potential of structured conversational data to enhance reasoning, especially in cases where domain-specific high-quality data is limited, paving the way for more effective and resource-efficient pretraining of LLMs.

Chapter 6

Scaling Self-Learning beyond Math Reasoning

Large Language Models (LLMs) have shown strong reasoning capabilities, particularly when enhanced through Reinforcement Learning (RL). While prior work has successfully applied RL to mathematical reasoning—where rules and correctness are well-defined—generalizing these methods to broader reasoning domains remains challenging due to limited data, the lack of verifiable reward structures, and diverse task requirements. In this chapter, we propose NEMOTRON-CROSTHINK, a framework that systematically incorporates multi-domain corpora, including both synthetic and real-world question-answer pairs, into RL training to improve generalization across diverse reasoning tasks. NEMOTRON-CROSTHINK addresses key challenges by (1) incorporating data from varied sources spanning STEM, humanities, social sciences, etc.; (2) applying structured templates (e.g., multiple-choice and open-ended) to control answer-space complexity; (3) filtering for verifiable answers; and (4) optimizing data blending strategies that utilizes data from multiple sources effectively.

6.1 Task Overview

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities across a wide range of tasks, with Reinforcement Learning (RL) playing a crucial role in refining their deep thinking abilities (Hu et al., 2025a; Aggarwal & Welleck, 2025; Luo et al., 2025a; DeepSeek-AI, 2025a; Qin et al., 2024; Huang et al., 2025; Team, 2025b). Recent advances in RL have been particularly successful in mathematical reasoning and coding, where well-defined rules and verifiable correctness criteria enable effective reward modeling. However, extending these techniques to broader reasoning domains presents significant challenges, such as—including limited training data for RL due to the difficulty of defining verifiable rewards, and ensuring generalization across diverse tasks.

Recent work (Hu et al., 2025a; Luo et al., 2025a; Cui et al., 2025) has shown a way to diversify RL training corpora by collecting datasets from multiple sources. However, they do not evaluate the relative importance of each source for downstream tasks, nor do they explore optimal data-blending strategies to maximize performance gains. Furthermore, prior research has largely focused on mathematical reasoning, overlooking the impact of incorporating non-math reasoning domains in RL-based learning for generalization in out-

of-distribution domains. A major challenge in applying RL to general-purpose reasoning tasks lies in designing a verifiable reward model for diverse answer spaces, as unlike mathematical reasoning—where correctness can be objectively verified—other reasoning tasks lack deterministic solutions. Moreover, reasoning process varies across domains and question types. For instance, mathematical problem-solving follows a rule-based, structured, and symbolic approach (Dehaene, 2011), whereas reasoning in fields such as law, physics, social sciences, and history often relies on narrative structures, contextual knowledge, and heuristic search strategies. Additionally, different question formats require distinct cognitive approaches — open-ended questions demand the generation of novel responses from scratch, while multiple-choice (MCQ) questions can often be solved more efficiently by evaluating the given options and selecting the most appropriate answer. Incorporating a diverse range of reasoning domains and question types into RL-based self-learning can enhance the broad reasoning capabilities of LLMs by exposing them to varied cognitive strategies and knowledge structures.

In this work, we propose NEMOTRON-CROSTHINK, a systematic way to incorporate multi-domain corpora for RL training that results in better generalization across a wide variety of tasks. As demonstrated in Figure 6.2, NEMOTRON-CROSTHINK comprises of phases that—(a) curate data from diverse sources, including synthetic data from raw web texts (CommonCrawl) and open-source question-answer pairs, spanning STEM, humanities, law, and social sciences (b) apply templates (MCQ/Open-Ended) to limit the answer-space for synthetically generated data (c) filter out samples that are infeasible for verifiable rewards (d) prepare blending recipes to combine different sources of data efficiently and finally (e) employ self-learning with RL to refine reasoning capabilities in diverse domains.

NEMOTRON-CROSTHINK demonstrates that integrating multi-domain data with different questions formats for RL significantly enhances reasoning ability of LLMs across diverse reasoning tasks. Notably, models trained with NEMOTRON-CROSTHINK not only achieve higher accuracy but also exhibit dynamic response strategies—generating concise answers for general-purpose questions and more detailed responses for math problems—thereby reducing inference cost while preserving task-specific rigor. In addition, NEMOTRON-CROSTHINK addresses the challenge of designing verifiable reward for non-deterministic domains by employing different templates on the curated data

to limit the nuances in the answer space diversity. This enables scalable, verifiable reward modeling for general purpose reasoning tasks, ensuring that RL-trained models generalize effectively across diverse benchmarks. Furthermore, NEMOTRON-CROSTHINK explores a simple yet effective filtering approach to rank general purpose reasoning data based on

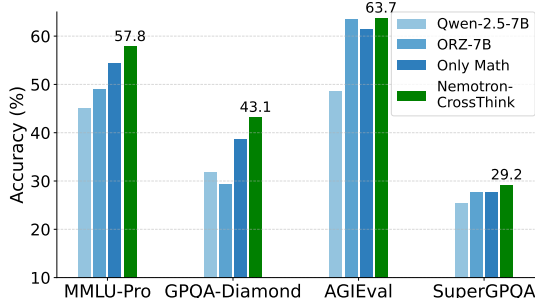


Figure 6.1: Employing self-learning with multi-domain data, NEMOTRON-CROSTHINK outperforms baseline models, including domain-specific training (Only Math) and Open-Reasoner-Zero (ORZ-7B), achieving consistent gains across all reasoning tasks.

complexity and shows that training with harder samples further amplifies the impact of RL across all domains.

6.2 Methodology

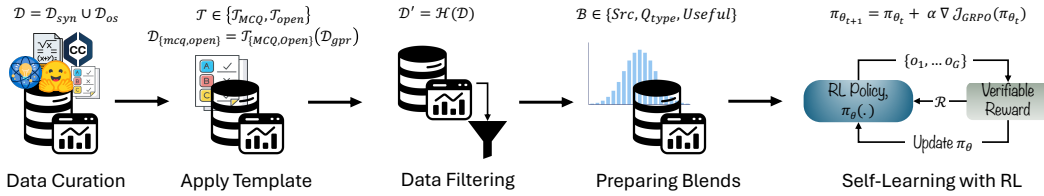


Figure 6.2: **NEMOTRON-CROSTHINK**. We (a) curate QA pairs from from synthetic (Common Crawl) and open-source datasets, categorized into general-purpose reasoning (\mathcal{D}_{gpr}) and mathematical reasoning (\mathcal{D}_{mr}); (b) apply structured templates to convert data into multiple-choice (MCQ) and open-ended formats, promoting diverse reasoning trajectories; (c) filter out unverifiable or ill-formatted responses; (d) train an RL policy using Group Relative Policy Optimization (GRPO). The final reward is used to update the policy, iteratively improving the model’s reasoning capabilities across diverse domains.

In this work, we investigate reasoning domains beyond mathematics and analyze the impact of RL on LLMs trained with datasets from diverse domains and question formats. A core pre-requisite for effective self-learning is access to high-quality, diverse, and reward-compatible training data (Xie et al., 2025b; Hu et al., 2025a). While mathematical reasoning has benefited from clean and verifiable datasets, extending RL to general-purpose reasoning domains remains underexplored due to the lack of structured, high-quality supervision. To address this, we explore methods for leveraging web documents and open-source QA benchmarks to collect general-purpose reasoning data. Incorporating a mix of structured and unstructured domains introduces a wide range of cognitive patterns and task-specific reasoning strategies which will further improve generalization. However, it introduces noise and ambiguity—particularly in open-ended formats—making it difficult to apply rule-based reward modeling reliably. To mitigate this, we apply task-specific templates to unify question and answer formats, limiting answer space variability and enabling simple but effective verifiable reward signals. Next, we apply a lightweight data filtering strategy to discard examples that are infeasible to verify—such as open-ended answers exceeding a certain length or MCQs with misaligned options—ensuring stable and interpretable RL training. Finally, we explore optimal data blending strategies that combine information across diverse domains and tasks. This allows us to investigate how the inclusion of general-purpose reasoning data complements mathematical reasoning, ultimately leading to broader and more adaptive generalization in LLMs.

Data Curation. We start with carefully curating datasets from multiple sources to ensure diversity in the training data. Our training dataset \mathcal{D} comprises two sources:

$$\mathcal{D} = \mathcal{D}_{syn} \cup \mathcal{D}_{os}$$

Here, $\mathcal{D}_{syn} \rightarrow$ synthetically generated data from Common Crawl (CC) and $\mathcal{D}_{os} \rightarrow$ publicly available open-source QA datasets. Each sources of data further consists of question answer

pairs related to general purpose reasoning and mathematics:

$$\mathcal{D}_{syn} \rightarrow \mathcal{D}_{syn_gpr} \cup \mathcal{D}_{syn_mr}; \quad \mathcal{D}_{os} \rightarrow \mathcal{D}_{os_gpr} \cup \mathcal{D}_{os_mr}$$

- **General Purpose Reasoning, \mathcal{D}_{gpr} :** We collect open source QA datasets (\mathcal{D}_{os_gpr})—Natural Reasoning (Yuan et al., 2025) and MMLU [Train] (Hendrycks et al., 2021b) that span multiple domains, including STEM fields (e.g., Physics, Computer Science), Economics, Social Sciences, and more. To enhance diversity, we further synthesize QA pairs from CC documents using the wide range of domains in MMLU as our seed domain. We denote this dataset as Syn-QA (\mathcal{D}_{syn_gpr}).

$$\mathcal{D}_{gpr} \rightarrow \mathcal{D}_{syn_gpr} \cup \mathcal{D}_{os_gpr}$$

- **Mathematical Reasoning, \mathcal{D}_{mr} :** As mathematical questions inherently require Chain-of-Thought derivations which emphasizes the LLM to think, we incorporate math reasoning corpus to our training data. We combine open-source mathematical reasoning datasets (\mathcal{D}_{os_mr}), such as MATH (Hendrycks et al., 2021e) and Numina-Math (Beeching et al., 2024). We generate additional math problems applying the similar technique as Ge et al. (2024) and define it as Persona-MATH (\mathcal{D}_{syn_mr}).

$$\mathcal{D}_{mr} \rightarrow \mathcal{D}_{syn_mr} \cup \mathcal{D}_{os_mr}$$

Applying Templates for Answer Space and Reasoning Diversity.

General purpose reasoning benchmarks are often divided into two categories: (a) Multiple Choice Questions (Hendrycks et al., 2021b; Wang et al., 2024d) and (b) Open-Ended Questions (Zhong et al., 2023). Recent works have ignored these variations in the answer space for consistent reward design across all tasks which are often predominantly math tasks (Hu et al., 2025a; Aggarwal & Welleck, 2025; Luo et al., 2025a). We hypothesize that each question type elicits different thinking patterns, leading to diverse reasoning trajectories in the model. Training on different question types will enhance the model’s ability to generalize by exposing it to diverse answer formats, thereby fostering different reasoning pathways.

Therefore, to observe the effect of question type in RL training, we synthesize \mathcal{D}_{gpr} using two templates: \mathcal{T}_{MCQ} - Multiple Choice Questions (MCQ), and \mathcal{T}_{Open} - Open-Ended questions. We convert the MCQ datasets (MMLU) to open-ended by removing the options from the questions.

$$\mathcal{D}_{mcq} = \mathcal{T}_{MCQ}(\mathcal{D}_{gpr}), \quad \mathcal{D}_{open} = \mathcal{T}_{Open}(\mathcal{D}_{gpr})$$

Additionally, some MCQ questions are incomplete without options (e.g., *Which of the following ways we can file taxes?*). We discard such questions to avoid confusion during answer generation. Finally, our general purpose reasoning data, \mathcal{D}_{gpr} , can be represented as:

$$\mathcal{D}_{gpr} = \mathcal{D}_{mcq} \cup \mathcal{D}_{open}$$

Data Source	Category	Type	Samples
MMLU [Train]	GPR	MCQ	99,842
Syn-QA	GPR	MCQ	192,930
NATURAL REASONING	GPR	OE	100,000
NuminaMath	MR	OE	87,350
PersonaSkill-MATH	MR	OE	100,000
Math	MR	OE	8523
Total			588,645

Table 6.1: Training data distribution by source and type. OE=Open-Ended; GPR=General-Purpose Reasoning; MR=Math Reasoning

Data Filtering and Formatting. To ensure high-quality training data, we apply a series of filtering and formatting steps, \mathcal{H} , to remove samples that are infeasible to evaluate with a simple rule-based reward function. Specifically, for \mathcal{D}_{mcq} , we check whether the correct answer appears within the question text itself. Given a question-answer pair (q, a^*) with answer choices $\{a_1, a_2, \dots, a_n\}$, we discard a sample if $a^* \notin \{a_1, a_2, \dots, a_n\}$.

For \mathcal{D}_{open} , such as those in the Natural Reasoning dataset, we discard samples that are challenging to evaluate with a rule-based reward function. Formally, we retain samples where $|w(a^*)| \leq 10$; $w(a^*)$ represents the number of words in the answer a^* .

Lastly, for the mathematical reasoning corpus, \mathcal{D}_{mr} , we remove entries that lack an associated answer, ensuring that all retained questions q have a valid response a^* , i.e., we discard samples where $a^* = \emptyset$.

$$\mathcal{D}' = \mathcal{H}(\mathcal{D}) = \left\{ (q, a^*, \{a_1, \dots, a_n\}) \in \mathcal{D} \mid \begin{array}{l} a^* \in \{a_1, \dots, a_n\} \quad (\mathcal{D}_{mcq}) \\ |w(a^*)| \leq 10 \quad (\mathcal{D}_{open}) \\ a^* \neq \emptyset \quad (\mathcal{D}_{mr}) \end{array} \right.$$

Data Blending. We study the impact of data diversity in three paradigms:

Category	Blend Name	Symbol	Blend Description
Data Source	Natural Distribution	\mathcal{B}_{nd}	Ratio of number of samples in a dataset divided by the total number of samples in all the datasets.
	More Math	$\mathcal{B}_{mr\uparrow}$	2:1 ratio of \mathcal{D}_{mr} and \mathcal{D}_{gpr}
	More General Purpose Reasoning	$\mathcal{B}_{gpr\uparrow}$	2:1 ratio of \mathcal{D}_{gpr} and \mathcal{D}_{mr}
Question Types	More MCQ	$\mathcal{B}_{mcq\uparrow}$	2:1 ratio of \mathcal{D}_{mcq} and \mathcal{D}_{open}
	More Open-Ended	$\mathcal{B}_{open\uparrow}$	2:1 ratio of \mathcal{D}_{open} and \mathcal{D}_{mcq}
Data Usefulness	Avg. Score	\mathcal{B}_{score}	Provide weight to each source based on their average benchmark performances

Table 6.2: **Overview of Data Blending Strategies.** Blends are categorized by data source, question type, and usefulness—each constructed to assess the impact of domain diversity, format variation, and task relevance on RL-based reasoning.

- **Data Source:** We have gathered questions from diverse domains including math (\mathcal{D}_{mr}), STEM, humanities, economics, history, law, social sciences, etc., (\mathcal{D}_{gpr}) and observe the effect of each source on RL training.
- **Question Types:** We investigate the impact of question types in downstream tasks.
- **Data Usefulness:** We further analyze what is the contribution of each data sources in downstream task performances. We initially run RL using individual data alone and then evaluate them across diverse downstream tasks. Based on their performances, we create a new blend.

Based on these three categories, we construct six distinct blends, summarized in [Table 6.2](#), with their corresponding dataset weight distributions detailed in [Table 12.24](#).

Reinforcement Learning with GRPO. We begin with a pretrained large language model (LLM) \mathcal{M} and a training blend \mathcal{B} , where each sample contains only the input prompt and the final answer which is verifiable. We employ Group Relative Policy Optimization (GRPO)

(Zhihong Shao, 2024). GRPO does not use a separate critic model and instead estimates the baseline from group scores, improving efficiency and reducing memory. For each question q , GRPO samples a group of outputs o_1, o_2, \dots, o_G from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_θ by maximizing the following objective:

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) &= \mathbb{E} \left[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q) \right] \\ &\times \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left[\min \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right) \right. \\ &\quad \left. - \beta D_{KL}(\pi_\theta \| \pi_{ref}) \right] \\ D_{KL}[\pi_\theta \| \pi_{ref}] &= \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1. \end{aligned} \quad (6.1)$$

where ϵ and β are hyperparameters, and $\hat{A}_{i,t}$ is the advantage, computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ corresponding to the outputs within each group:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}$$

Rule Based Reward Modeling. To guide the reinforcement learning process, we employ a rule-based reward system designed for verifiable evaluation. Similar to (DeepSeek-AI, 2025a), we define the total reward function \mathcal{R} as the logical and of an accuracy reward \mathcal{R}_{acc} and a format reward \mathcal{R}_{format} :

$$\mathcal{R} = \mathcal{R}_{acc} \wedge \mathcal{R}_{format}.$$

This implies that the output will get reward only when both the answer and the format are correct.

Accuracy Reward: The accuracy reward evaluates correctness based on whether the model’s response p is similar to the ground truth solution a to satisfy the correctness criteria:

$$\mathcal{R}_{acc}(p, a) = \begin{cases} 1, & \text{if equal}(p, a), \\ 0, & \text{otherwise.} \end{cases}$$

Format Reward: The format reward ensures the response a is structured according to predefined tags, where the reasoning will reside in ‘<think></think>’ tokens and the final answer will be shown inside `\boxed{\}`:

$$R_{format}(a) = \begin{cases} 1, & \text{if } F(a), \\ 0, & \text{otherwise.} \end{cases}$$

where $F(a)$ returns True if a is correctly formatted and False otherwise.

6.3 Experiments and Results

6.3.1 Experimental Setup

Training Details. We adopt Qwen2.5-7B and Qwen2.5-32B (Team, 2024a) as our baseline models, \mathcal{M} , which demonstrate strong generalization capabilities across various natural lan-

guage reasoning tasks. We directly apply GRPO training on \mathcal{M} using the veRL framework¹, which is an open-source implementation of the HybridFlow RLHF framework (Sheng et al., 2024). We train the base models with key settings including a constant learning rate of 1e-6, a batch size and PPO mini batch size of 128 and a maximum context length of 5000 tokens. Each generation step contains 128 unique prompts sampled from the dataset, and performing 8 rollouts with temperature and top-p both set to 1.0. We set KL coefficient to 0.001 in all experiments. During training, the model is directly exposed to mixed types of questions from different domains. Note that we did not conduct extensive hyperparameter tuning, so one can expect further improvements with additional optimization.

Evaluation Metrics. To comprehensively evaluate our models’ reasoning capabilities, we conduct experiments on diverse benchmarks spanning mathematical and general purpose reasoning. We evaluate our models on MATH-500(Hendrycks et al., 2021e), AMC23, test set of MMLU (Hendrycks et al., 2021b), MMLU-PRO (Wang et al., 2024d), AGIEVAL (Zhong et al., 2023), GPQA-DIAMOND (Rein et al., 2024) and SUPERGPQA (Team et al., 2025). Notably, SUPERGPQA is a recent and rigorous benchmark designed to test the generalizability of LLMs across 285 graduate-level disciplines, including underrepresented domains like industry, agriculture, and service-related fields. Unlike existing benchmarks that concentrate on well-represented domains (e.g., math, law, physics), SUPERGPQA captures long-tail knowledge and includes a wide range of real-world professional disciplines, making it a reliable and discriminative frontier for evaluating generalizability in LLMs. For both open-ended and MCQ questions, we check the final answer inside the `\boxed{}` format and compare with the ground truth solution. For MCQ benchmarks (e.g., MMLU, GPQA-DIAMOND, etc.), we format the ground truth in the test set to contain both the correct option and the option description to make it consistent with our training data. For each benchmark, we report accuracy averaged over 3 independent inference runs using greedy decoding.

6.3.2 Results

Analyze the effect of Individual Datasets. To prepare an effective blend using diverse sources of data, we begin by understanding impact of individual data sources on the self-learning paradigm so that we can prioritize the useful data sources and provide less weights to others. In this setup, we employ self-learning using \mathcal{M} =Qwen-2.5-7B and taking each dataset separately. To make consistent comparison across different data sources, we keep the training recipe constant for all experiments. We run a controlled experiments and train each models for fewer steps (250 steps) and evaluate them on the last checkpoint.

Table 2 shows that different datasets have varying impacts on downstream accuracies across reasoning benchmarks. Notably, the NuminaMath yields the highest overall average, outperforming the baseline (\mathcal{M}) by over 8.30%. Its strength is especially pronounced on mathematical tasks such as MATH-500and AMC23 but additionally it achieves superior accuracies on general purpose reasoning tasks showing a strong generalization across diverse domains. The Syn-QA dataset demonstrates a \sim 1.0% improvement over baseline with stronger accuracy in MMLU-PRO, AGIEVAL and MATH-500tasks, suggesting that synthetically generated instruction-style data can generalize well when aligned with benchmark distributions. Natural Reasoning, despite modest scores on language-rich benchmarks, de-

¹<https://github.com/volcengine/verl>

Data Source	MMLU	MMLU-PRO	GPQA-DIAMOND	AGIEVAL	SUPERGPQA	MATH-500	AMC23	Avg
\mathcal{M}	74.20	45.00	31.82	48.59	25.36	48.30	40.00	44.75
MMLU [Train]	69.76	38.50	32.83	47.66	27.69	22.00	5.00	34.78
Syn-QA	70.45	52.41	30.81	52.10	24.57	54.20	35.00	45.65
Natural Reasoning	68.89	31.33	33.33	46.65	22.44	68.60	42.50	44.82
NuminaMath	72.94	52.05	33.84	54.39	26.97	76.20	55.00	53.06
PersonaSkill-Math	53.99	28.08	18.69	45.69	16.92	77.20	50.00	41.51
Math	63.30	31.64	21.72	51.95	18.31	78.40	50.00	45.04

Table 6.3: **Results of Self-Learning on Individual Datasets.** Each row shows the downstream evaluation results after self-learning on a single data source. Results highlight the varying strengths of individual datasets across general-purpose and mathematical benchmarks.

livers a surprisingly strong overall average, driven by high scores in MATH-500 and AMC23. This indicates that reasoning-focused datasets, even if less curated, can contribute meaningfully in math-adjacent tasks. On the other hand, Persona-Math, although strong in math, suffers from low generalization across most benchmarks. Finally, the MMLU [Train] dataset underperforms across most tasks, specifically in math reasoning domains, suggesting that self-learning with raw MMLU [Train] data alone is insufficient for generalization. However, it obtains the best score for SUPERGPQA, which requires reasoning across wide range of cross-disciplinary domains. This highlights the potential of MMLU [Train] in capturing broad conceptual knowledge and supporting transfer to long-tail domains, making it a valuable component when targeting general-purpose reasoning benchmarks. While preparing blends for *Data Usefulness*, we use the average accuracies of individual sources to obtain \mathcal{B}_{score} i.e., we provide more weight to datasets like Syn-QA, NuminaMath and less to MMLU [Train].

Analysis across Blends. We observe the effect of NEMOTRON-CROSTHINK in three different categories using six different blends. To show the distinction between natural distribution and selective weighting of domains, we also prepare \mathcal{B}_{nd} , which represents data sampled in proportion to each dataset’s original size. Additionally, to analyze the impact of within-domain training versus cross-domain blending, we introduce a separate category called Single Source. We prepare two domain-specific blends: \mathcal{B}_{only_mr} , using only \mathcal{D}_{mr} data, and \mathcal{B}_{only_gpr} , using only \mathcal{D}_{gpr} data. We further compare NEMOTRON-CROSTHINK with a recent math-centric self-learning approach, OPEN-REASONER-ZERO (ORZ) (Hu et al., 2025a)—which achieved superior accuracy in math benchmarks by training RL on combination of math data. For fair comparison we evaluate the 7B model using our eval setup.

As shown in Table 6.4, each blending strategy consistently outperforms the base model, \mathcal{M} , by a significant margin. The natural distribution blend, \mathcal{B}_{nd} , yields a notable improvement of over 13% on average compared to \mathcal{M} , suggesting that simply increasing the amount of training data—even without rebalancing—can be beneficial.

$\mathcal{B}_{gpr\uparrow}$ from the *Data Source* category achieves the highest overall average, as well as the strongest results across most reasoning-focused benchmarks (e.g., +12.82% on MMLU-PRO and +15.12% on AGIEVAL). Notably, it performs relatively $\sim 5\%$ on average better than ORZ. While \mathcal{B}_{only_math} performs slightly better on math-specific tasks, such as a marginal 1% gain on MATH-500, it lags behind on non-math reasoning benchmarks—underperforming $\mathcal{B}_{gpr\uparrow}$

Model	Category	Blend	MMLU	MMLU-PRO	GPQA-DIAMOND	AGIEVAL	SUPERGPQA	MATH-500	AMC23	Avg
\mathcal{M}			74.20	45.00	31.82	48.59	25.36	48.30	40.00	44.75
ORZ			73.20	48.90	29.30	63.49	27.60	81.40	62.50	55.20
*CROSSLINK		\mathcal{B}_{nd}	73.18	54.81	38.07	59.99	26.54	77.00	60.00	55.66
	Data Source	$\mathcal{B}_{mr}\uparrow$	74.85	55.51	40.10	61.47	26.81	77.80	67.50	57.72
		$\mathcal{B}_{gpr}\uparrow$	74.94	57.82	38.58	63.71	29.16	77.60	65.00	58.12
	Question Types	$\mathcal{B}_{mcq}\uparrow$	74.26	55.77	39.59	62.54	28.05	78.00	60.00	56.89
		$\mathcal{B}_{open}\uparrow$	74.46	55.82	43.15	61.28	26.82	78.40	62.50	57.49
	Data Usefulness	\mathcal{B}_{score}	74.70	56.16	40.10	59.80	27.37	78.00	62.50	56.95
	Single Source	\mathcal{B}_{only_mr}	74.24	54.26	38.58	61.39	27.69	78.60	70.00	57.82
\mathcal{B}_{only_gpr}		72.77	52.06	37.06	56.56	27.44	72.20	55.00	53.30	

Table 6.4: **Results of NEMOTRON-CROSSLINK-7B across Blends.** Multi-domain blend $\mathcal{B}_{gpr}\uparrow$ achieves the highest overall average accuracy, outperforming domain-specific and naturally sampled blends—underscoring the benefit of self-learning with diverse reasoning data. (*) Due to the space shortage, we use *CROSSLINK to refer NEMOTRON-CROSSLINK.

by $\sim 3\text{--}4\%$ on tasks like AGIEVAL, SUPERGPQA, and MMLU-PRO. The same trend is also seen with ORZ. To better understand these differences, we analyze sub-category accuracies in [section 12.14](#) and find that $\mathcal{B}_{gpr}\uparrow$ shows large relative gains in non-math categories while differences in math subcategories are either negligible or even favor $\mathcal{B}_{gpr}\uparrow$ in some tasks. This highlights that general-purpose reasoning data offers strong cross-domain transfer with minimal compromise on math accuracy, making it more versatile.

Both $\mathcal{B}_{mcq}\uparrow$ and $\mathcal{B}_{open}\uparrow$ in *Question Types* category show consistent gains, with the latter achieving a slight edge (0.6% improvement on average). In addition, $\mathcal{B}_{open}\uparrow$ yields stronger results on mathematical benchmarks. Mathematical problems are inherently open-ended in structure. As a result, highlighting more open-ended domains aligns with the format and reasoning demands of math tasks. This suggests that diversity in question formats—especially open-ended reasoning—can better generalize to both general purpose reasoning and math-focused downstream tasks.

Regarding *Data Usefulness*, the score-based selection strategy (\mathcal{B}_{score}) outperforms the base model \mathcal{M} , indicating the effectiveness of selective data curation. However, despite focusing more on the better performing datasets in [Table 6.3](#), \mathcal{B}_{score} is overall worse than blends like $\mathcal{B}_{mr}\uparrow$ or \mathcal{B}_{only_math} . This gap arises because \mathcal{B}_{score} assigns weights based solely on average dataset scores, without accounting for task-specific strengths. For instance, Math and Persona-Math receive higher weights than Natural Reasoning or MMLU due to their math accuracy, despite the latter performing significantly better on general-purpose reasoning tasks. In contrast, domain-aware blends selectively prioritize datasets based on their utility within specific domains, leading to more effective coverage and stronger scores across both math and general-purpose reasoning tasks.

To investigate the impact of single-domain versus mixed-domain training data in RL, we compare the *Single Source* category with other blending strategies. Notably, \mathcal{B}_{only_mr} achieves the highest average math score (56.20%) among all blends, ranking as the second-best blend overall in terms of average accuracy. In contrast, while \mathcal{B}_{only_gpr} outperforms the base model \mathcal{M} , it underperforms in mathematical reasoning tasks. Surprisingly, despite being tailored for general-purpose reasoning, \mathcal{B}_{only_gpr} also lags behind \mathcal{B}_{only_mr} by 4.2% on average across non-math reasoning benchmarks. This counterintuitive finding suggests that to

obtain maximum gain in general purpose reasoning tasks we need to include mathematical problems in the training blend. As discussed earlier, $\mathcal{B}_{gpr\uparrow}$ gets the best average reasoning accuracy which consists of both math and general purpose reasoning datasets. This confirms that math data alone is transferable to structured reasoning tasks, whereas general-purpose data is less effective when isolated.

6.4 Ablations

NEMOTRON-CROSTHINK is more token efficient in responses compared to \mathcal{B}_{only_mr} . To further understand the influence of multi-domain data in response generation, we compare the average token lengths of correct and incorrect responses between models trained on two blends: $\mathcal{B}_{gpr\uparrow}$ and \mathcal{B}_{only_mr} . As shown in Figure 6.3, on general-purpose reasoning (GPR) benchmarks, $\mathcal{B}_{gpr\uparrow}$ consistently outperforms \mathcal{B}_{only_mr} and ORZ (Hu et al., 2025a), not only in accuracy (as shown in Table 6.4) but also in response efficiency—producing correct answers with significantly fewer tokens². For instance, on MMLU, the average token count for correct responses is 229 for $\mathcal{B}_{gpr\uparrow}$, compared to 351 for \mathcal{B}_{only_mr} . This demonstrates that exposure to multi-domain data enables the model to internalize a more efficient reasoning strategy, leading to both improved performance and reduced inference cost.

In contrast, on math-specific benchmarks, \mathcal{B}_{only_mr} and ORZ perform slightly better in accuracy, as expected due to domain alignment. Interestingly, correct responses are generally longer than in reasoning tasks as solving math problems inherently requires detailed, multi-step derivations, hypothesis exploration, verification and refinement. Despite this, the $\mathcal{B}_{gpr\uparrow}$ shows its adaptability by generating longer responses for math tasks and shorter ones for GPR tasks—indicating a dynamic response strategy learned through multi-domain training. As shown in Table 12.25, $\mathcal{B}_{gpr\uparrow}$ has a wide dynamic range for generating responses. It increases its average tokens by 62% when generating responses for math tasks (Mean Tokens=622) as opposed to general reasoning tasks (Mean Tokens=385). Whereas, \mathcal{B}_{only_mr} increases its average tokens only by 14% (Mean Tokens=731 for math tasks and Mean Tokens=639 for general reasoning tasks) showing a much smaller dynamic range. This trend is also mirrored in ORZ, trained on a high-quality blend of math datasets, which shows an even smaller increase (12%) in average token length across domains.

This adaptive behavior highlights a key strength of multi-domain training: it equips the model with the flexibility to tailor its response style to the nature of the task. By learning from a diverse range of domains, $\mathcal{B}_{gpr\uparrow}$ learns to reason efficiently—across all tasks, $\mathcal{B}_{gpr\uparrow}$ uses on average 28% fewer tokens for correct responses than \mathcal{B}_{only_mr} —producing compact yet accurate answers where appropriate, and detailed ones when necessary.

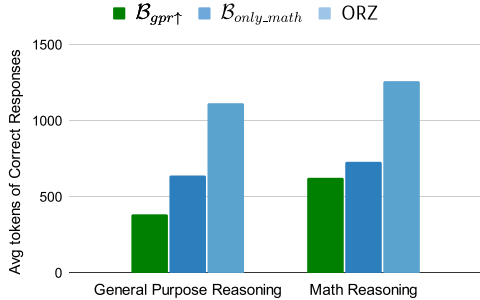


Figure 6.3: Token efficiency comparison of models trained on $\mathcal{B}_{gpr\uparrow}$ (multi-domain blend) and two single domain blends (\mathcal{B}_{only_math} and ORZ).

²Detailed categorization per task is shown in Appendix 12.13.

Data Format Study: Question and Answer Templates. To better understand how training data formatting affects model performance, we conduct two controlled studies focused on question and answer template design, as shown in Table 6.5 and Table 6.6.

In Table 6.4, we observe that $\mathcal{B}_{open\uparrow}$ outperforms $\mathcal{B}_{mcq\uparrow}$, suggesting that models trained on more open-ended data generalize better across benchmarks. This motivated us to investigate whether converting all questions into a unified open-ended format leads to better performance. In *Question Template Study*, we use the natural distribution blend (\mathcal{B}_{nd}) and only perturb the question template. To generate the open-ended variant, we remove the answer options from MCQs, prompting the model to produce an answer without selecting from predefined choices.

Question Type	MMLU	MMLU-PRO	GPQA-DIAMOND	AGIEVAL	SUPERGPQA	MATH-500	AMC23	Avg
MCQ + OPEN-ENDED	73.18	54.81	38.07	59.99	26.54	77.00	60.00	55.66
OPEN-ENDED	74.61	54.36	39.09	59.30	29.16	76.60	65.00	56.87

Table 6.5: **Impact of Question Format.** Converting all questions to open-ended format improves accuracy across benchmarks, reducing reliance on option guessing and encouraging deeper reasoning.

Table 6.5 illustrates that the open-ended-only configuration consistently outperforms the mixed-format setting across nearly all benchmarks, achieving 1.21% higher average score. Notably, it leads to significant improvements on reasoning-intensive and traditionally MCQ-formatted benchmarks such as MMLU, SUPERGPQA, and GPQA-DIAMOND. This result may be attributed to the inherent structure of MCQ questions, where random guessing can yield an accuracy of approximately 25% in MMLU and GPQA-DIAMOND benchmarks where we have only four options. In contrast, open-ended questions eliminate this guessing advantage, compelling the model to rely more heavily on reasoning to arrive at a correct answer. By reducing the likelihood of reward hacking through random option selection, the open-ended format encourages more robust reasoning and leads to improved generalization.

In the *Answer Template Study*, we investigate how the format of output labels influences training effectiveness on MCQ-style datasets. We compare two answer templates: Long - the model is trained to generate both the option label and its corresponding description (e.g., (A) The sky is blue), and Short - the model is trained to output only the option label (e.g., A). For this study, we use the \mathcal{B}_{only_gpr} blend, which primarily consists of MCQ datasets (Table 6.1), making it ideal for analyzing the effects of answer formatting in this setting.

Answer Type	MMLU	MMLU-PRO	GPQA-DIAMOND	AGIEVAL	SUPERGPQA	MATH-500	AMC23	Avg
Long	72.77	52.06	37.06	56.56	27.44	72.20	55.00	53.30
Short	74.22	54.56	39.59	58.01	28.39	74.20	52.50	54.50

Table 6.6: **Impact of Answer Format.** Using short-form answers improves accuracy by reducing output ambiguity and avoiding penalization from rigid reward functions in rule-based training.

Table 6.6 shows that the short-form answer template consistently outperforms the long-form variant, with a 1.20% improvement in average accuracy. This trend holds across both reasoning and mathematical benchmarks. These results suggest that reducing the

complexity of the output space helps minimize ambiguity and allows the model to better align its predictions with the structure of the question. Furthermore, when training with long-form answers using a rule-based reward (e.g., exact string matching), the model is frequently penalized for minor deviations in phrasing, even when the correct option is selected. For instance, if the model outputs the correct option label but paraphrases the description slightly, the strict reward signal treats it as incorrect. This introduces noisy supervision and may hinder learning. While this issue could be mitigated by designing a more flexible reward function (e.g., based on semantic similarity or option-label matching), our goal in this work is to keep the approach simple and interpretable. As such, we adopt a naive rule-based reward for clarity and reproducibility, and leave more sophisticated reward designs for future investigation.

Difficulty Filtering. Training with high-quality data is a key factor in self-learning to ensure efficient and stable learning and to obtain correct reward signals. Recent works (Hu et al., 2025a; Luo et al., 2025a; Cui et al., 2025) have explored various filtering strategies to remove noisy reference answers from datasets, focusing on data that is easily verifiable using simple rule-based rewards. Zeng et al. (2025) further investigate data selection based on question complexity, showing that as the difficulty of the training data increases, the resulting model achieves better downstream accuracy. However, their approach relies on datasets like MATH-500 that come with predefined difficulty scores. In this work, we explore a simple approach to estimate question difficulty for general purpose reasoning datasets that do not come with explicit difficulty labels. Specifically, we label questions as ‘difficult’ if they are answered incorrectly by a smaller model (Qwen-2.5-7B) in a zero-shot setting and filter out the ‘easy’ questions. The intuition is that questions easily answered by a base model are likely to be knowledge-based or shallow in reasoning depth, whereas those it fails on are likely to require deeper reasoning or broader generalization. We construct two versions of our training dataset $\mathcal{B}_{gpr\uparrow}$ —an unfiltered set containing all questions, and a filtered set ($\mathcal{B}_{f(gpr)\uparrow}$) that retains only the difficult samples—and use them to train separate instances of a larger model $\mathcal{M} = \text{Qwen-2.5-32B}$.

Model	Blend	MMLU	MMLU-PRO	GPQA-DIAMOND	AGIEVAL	SUPERGPQA	MATH-500	AMC23	Avg
Qwen-2.5-32B		83.30	55.10	40.40	62.77	33.16	60.55	45.00	54.33
NEMOTRON-CROSTHINK-32B	$\mathcal{B}_{gpr\uparrow}$	83.57	68.83	46.70	73.90	37.99	82.40	67.50	65.84
	$\mathcal{B}_{f(gpr)\uparrow}$	83.60	69.43	49.75	75.82	38.34	84.00	75.00	67.99

Table 6.7: **Difficulty-Based Filtering.** Filtering $\mathcal{B}_{gpr\uparrow}$ to retain only hard examples ($\mathcal{B}_{f(gpr)\uparrow}$) yields consistent gains across all tasks, highlighting the effectiveness of selective training on challenging data.

According to Table 6.7, this filtering approach results in consistent performance improvements across all evaluated benchmarks. While both filtered and unfiltered models outperform the original baseline Qwen-2.5-32B, the model trained on the filtered dataset—denoted as $\mathcal{B}_{gpr\uparrow}f$ —achieves the highest accuracy on every task. The gains are especially prominent in complex benchmarks such as MMLU-PRO, GPQA-DIAMOND, AGIEVAL, and AMC23, where the filtered model improves by up to 2–8% over its unfiltered counterpart. On average, filtering boosts overall accuracy by 2.15%, a notable gain considering that it comes from training on fewer but harder examples. This suggests that selectively training on challenging

examples can yield more robust and generalizable models, likely due to stronger gradient signals and a focus on harder-to-learn reasoning patterns.

6.5 Related Work

Evolution of Reasoning in LLM. Large Language Models have demonstrated remarkable dominance across numerous Natural Language Processing tasks. To enhance the complex reasoning capabilities of LLMs, (Wei et al., 2022) introduce Chain-of-Thought (CoT), which incorporates multi-step intermediate reasoning before arriving at final conclusions. CoT exhibits significant advantages across multiple domains, including mathematics, science, and programming. Subsequently, (OpenAI, 2024a) further explore CoT and propose the Long Chain-of-Thought framework. In Long CoT, LLMs demonstrate advanced cognitive behaviors such as reflection, verification, correction, and multipath exploration, thereby further enhancing their problem-solving capabilities in complex reasoning tasks. Moreover, Long CoT exhibits excellent test-time scaling properties, where increased computational resources correlate with improved reasoning outcomes. Models like QwQ (Team, 2024b; 2025b), DeepSeek-R1 (DeepSeek-AI, 2025a), Kimi k1.5 (Team, 2025a), and InternThinker (Cai et al., 2024) have successfully experimented with Long CoT for enhanced reasoning, combining fine-tuning and Reinforcement Learning to elevate the performance of open-source reasoning models to unprecedented levels. Notably, subsequent models such as Open-Reasoner-Zero (Hu et al., 2025a), Open-R1 (Face, 2025), O1-Replication (Qin et al., 2024; Huang et al., 2024c; 2025), s1 (Muennighoff et al., 2025) and LIMO (Ye et al., 2025) observes significant benefits from Long CoT even in smaller models through simple distillation.

Self-Learning beyond Math. High-quality training data are crucial for scalable Reasoner-Zero training. Most of the recent works emphasize mathematical benchmark-centric data (AMC, AIME, Math, Olympiads, and AoPS) for reinforcement learning (Hu et al., 2025a; Aggarwal & Welleck, 2025; Trung et al., 2024; Ye et al., 2025; Zeng et al., 2025) as designing verifiable rewards is much easier for math tasks. They exclude problems such as multiple choice and proof-oriented problems which reduces the answer space diversity. MCQ type of questions are important for MMLU and other non-reasoning centric tasks. For a rule-based reward model, the format of input data and the final answer is crucial and largely underexplored. Furthermore, their additional sources of data synthesis approach has no details making it infeasible to scale for domains other than math. The kind of data and the ratio of each type of data important for the overall improvement of LLMs across multiple benchmarks have yet to be explored.

Data Sampling in RL. Recent works have widely explored the idea of combining data from multiple sources during RL training to enhance the diversity of reasoning tasks and improve model generalization (Hu et al., 2025a; Luo et al., 2025a; Zeng et al., 2025; Wen et al., 2025). These studies primarily concentrate on the mathematical domain, where rule-based correctness allows for straightforward reward modeling. In such setups, data sampling strategies are often driven by factors like question complexity or the ease with which answers can be verified algorithmically. For instance, questions are filtered or prioritized based on whether they are solvable with deterministic programs or satisfy certain symbolic constraints. A notable direction is curriculum learning, where Xie et al. (2025b) utilizes synthetically generated puzzle-like data from Xie et al. (2025a) to control the difficulty level

and study the progression of learning. However, these works remain narrowly focused on highly structured domains such as logic puzzles or math word problems. [Yeo et al. \(2025\)](#) has shown that including 50% of math and 50% of noisy verifiable data from WebInstruct-462k [Yue et al. \(2024\)](#) yields best MMLU-PRO score in RL setup—indicating the potential of mixing of domains in the training blend. However, it is unclear how this benefit is attributed to the inclusion of non-math reasoning data as 68.36% of WebInstruct-462k is about math. They have performed filtering to obtain data with feasible verifiable reward and this is used to boost and prioritize the mathematical domain over other domains. Despite this progress, there is a lack of systematic investigation into how including non-math reasoning data—such as legal analysis, social science, commonsense inference, or historical interpretation—affects RL training. NEMOTRON-CROSTHINK is the first systematic framework to incorporate multi-domain and multi-format data into RL, introducing verifiable reward mechanisms for non-deterministic domains and demonstrating that blending diverse reasoning sources leads to stronger generalization across benchmarks.

6.6 Summary

We present NEMOTRON-CROSTHINK, a simple and scalable framework for improving the generalization abilities of LLMs through reinforcement learning with multi-domain corpora. By combining data from diverse reasoning domains and applying lightweight filtering and formatting strategies, NEMOTRON-CROSTHINK enables consistent gains across both general-purpose and mathematical benchmarks. Our best-performing blend—constructed with a 2:1 ratio of general-purpose to math data—achieves a 13.36% average improvement over strong baselines, with gains further amplified by difficulty-based filtering and thoughtful template design. Preparing blends admits many designs beyond this single recipe: as detailed in our methodology, mixtures can be structured along multiple axes—balancing data sources, varying the proportion of multiple-choice versus open-ended items, and reweighting sources by downstream usefulness—and we only instantiate a small, interpretable slice of this broader space; systematically exploring alternative blend constructions, schedules, and weighting schemes is important future work. Importantly, these benefits persist across model scales and task types, demonstrating that data diversity, not just data volume, is key to broader reasoning capabilities. NEMOTRON-CROSTHINK offers a practical recipe for building more generalizable, efficient, and reliable LLMs under the RL paradigm—paving the way for scalable self-learning beyond math.

Part II

Effect of reasoning data across LLM development phases

Chapter 7

Synergy between Pretraining and Post-Training Data

Training phases for Large Language Models (LLM) broadly consists of two consecutive phases: Pretraining and Post-Training. Pretraining an LLM involves training a neural network on a massive corpus of text data, which includes books, articles, and websites, to help it understand the nuances of human language. This initial phase of training sets the foundation for the model’s ability to parse syntax, grasp semantics, and generate text that is contextually relevant and grammatically correct. Depending on the quality and characteristics of the corpora, current literature follow 2-3 stage procedure to pretrain an LLM where the initial phase is generally followed by a phase with a pre-training based on high-quality corpora (Grattafiori et al., 2024; Feng et al., 2024b; OLMo, 2025; Blakeney et al., 2024; DeepSeek-AI, 2025b). In contrast, LLM post-training phases occur after the initial pretraining of a large language model (LLM) on a vast corpus of text, which further extends to supervised finetuning (SFT) and reinforcement learning (Direct Preference Optimization (Rafailov et al., 2023), Proximal Policy Optimization (Schulman et al., 2017)) (Grattafiori et al., 2024). These phases are crucial for refining the model’s capabilities, aligning it with human preferences, and optimizing its real-world performance.

In the recent literature, introducing long thought data in post-training has demonstrated significant breakthroughs in highly complex reasoning tasks, such as mathematical competitions (Hendrycks et al., 2021e), PhD-level scientific QA (Rein et al., 2024; Phan et al., 2025), and software engineering (Jimenez et al., 2024). However, there has been little to no explorations on how effective these high quality long thought data is during pretraining. The state-of-the-art proprietary LLMs being opaque about their pretraining corpora and overall the exploration with pretraining being highly expensive compared to post-training have become the major stimulation for the community to solely focus on post-training.

In this part, we shift our focus to the synergy between long chain-of-thought (CoT) reasoning QA data (i.e., SFT data) in pretraining and post-training. Our central goal is to determine the ideal balance of such reasoning data across the two phases in order to maximize downstream improvements after reinforcement learning.

We hypothesize that introducing reasoning tasks early in pretraining can foster intrinsic cognitive capabilities, leading to stronger generalization and adaptability during downstream

tasks. A key distinction, however, is that pretraining requires orders of magnitude more diverse data than post-training, which raises important challenges for scaling reasoning data beyond mathematics—where most existing resources lie—into broader domains such as science, law, puzzles, and deductive reasoning, each with distinct reasoning patterns.

7.1 Task Overview

The reasoning abilities of Large Language Models (LLMs) have advanced considerably, with post-training on reasoning data driving significant breakthroughs in reasoning tasks, such as math competitions (Hendrycks et al., 2021e), PhD-level scientific QA (Rein et al., 2024; Phan et al., 2025), and software engineering (Jimenez et al., 2024). This progress has been largely driven by mid- or post-training LLMs on high-quality, reasoning-intensive datasets—often featuring long chain-of-thought (CoT) examples (Guha et al., 2025; Moshkov et al., 2025; Zhou et al., 2025a; Gandhi et al., 2025b; Wang et al., 2025). While this approach has proven effective, it treats reasoning as a specialized skill to be layered onto a generalist base. In addition, the impact of incorporating reasoning data during pretraining—and the potential synergistic effects on subsequent post-training—remains a critical yet less explored frontier. This research gap persists due to the prohibitive computational cost of end-to-end pretraining experiments and the opacity surrounding proprietary training recipes, which has concentrated community efforts on the more accessible post-training phase.

The synergy between post-training phases has been widely explored (Liu et al., 2025c; Chen et al., 2025b; Chu et al., 2025), yet conclusions vary with training data and scale, and their applicability to pretraining remains vague in the current literature. In this work, we investigate not just *which* reasoning data, but *when* to train with such reasoning data by studying the synergy between pretraining and post-training. Our central goal is to determine the ideal balance of such reasoning data across the two phases in order to maximize downstream accuracies after reinforcement learning. This motivates the following research questions:

- *Is a reasoning-rich pretraining essential, or can a model “catch up”?* We investigate whether a model pretrained without reasoning data can match the performance of its reasoning-aware counterparts by simply undergoing a more intensive SFT phase.
- *Does inclusion of reasoning data make the base LLM overfitted and less generalizable to sustain gains in subsequent training phases?* While recent literature highlights overspecialization of reasoning during post-training can be detrimental (Gupta et al., 2025; Luo et al., 2025b), investigations of this effect in pretraining remain limited.
- *Does data diversity in pretraining impact stability and specialization during SFT?* Specifically, does using the same reasoning data in both pretraining and SFT lead to robust skill mastery, or does a *narrow pretraining focus* risk catastrophic forgetting when the model is later fine-tuned?
- *Does the complexity and quality of reasoning data matter when incorporated during pretraining of the base model?* Current literature explores this mostly from SFT stage (Zhou et al., 2023; Guha et al., 2025), making it obscure whether difficulty or noisiness in the early phase of training directly impacts reasoning development or not.

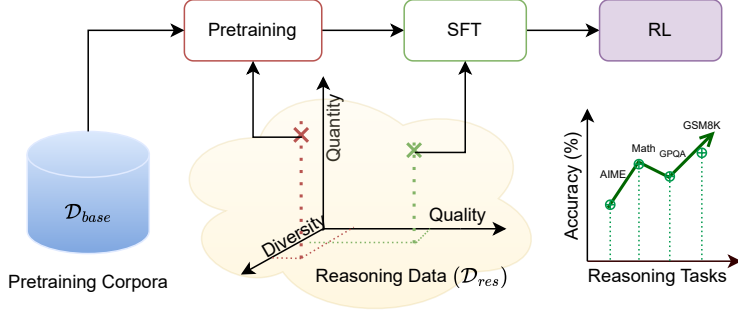


Figure 7.1: We systematically inject reasoning-style data (\mathcal{D}_{res}) at different phases of training—pretraining versus SFT—while varying its *diversity*, *quantity*, and *quality*. Our results show an asymmetric principle: diversity and scale matter most during pretraining, whereas quality dominates in SFT. This allocation strategy compounds through reinforcement learning (RL), yielding sustained gains across complex reasoning benchmarks.

7.2 Methodology

Our methodology is designed to systematically determine the optimal strategy for allocating reasoning data between the pretraining and supervised fine-tuning stages of LLM development. We frame this as an optimization problem where the goal is to maximize the final model’s downstream accuracies, \mathcal{P} . This is a function of the reasoning data introduced during pretraining, $\mathcal{D}_{\text{res}}^{\text{PT}}$, and the data used for supervised fine-tuning, $\mathcal{D}_{\text{res}}^{\text{SFT}}$. Our objective is to find the optimal data configurations, $(\mathcal{D}_{\text{res}}^{\text{PT}*}, \mathcal{D}_{\text{res}}^{\text{SFT}*})$, that solves the following:

$$(\mathcal{D}_{\text{res}}^{\text{PT}*}, \mathcal{D}_{\text{res}}^{\text{SFT}*}) = \arg \max_{\mathcal{D}_{\text{res}}^{\text{PT}}, \mathcal{D}_{\text{res}}^{\text{SFT}}} \mathcal{P}(\theta_{\text{final}})$$

where θ_{final} represents the parameters of the final model trained on data recipes defined by the choice of reasoning data at both stages.

Let $\mathcal{D}_{\text{base}}$ denote the general pretraining corpus and we define a model $\mathcal{M}(\theta)$ with parameters θ trained in two stages:

$$\begin{aligned} \text{Pretraining: } \theta_{\text{PT}} &= \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{base}} \cup \mathcal{D}_{\text{res}}^{\text{PT}}} \mathcal{L}_{\text{LM}}(f_{\theta}(x), y), \\ \text{SFT: } \theta_{\text{SFT}} &= \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{res}}^{\text{SFT}}} \mathcal{L}_{\text{SFT}}(f_{\theta}(x), y), \end{aligned}$$

Evaluation Objective. The central research question can be expressed as analyzing the function:

$$\mathcal{P}(\mathcal{D}_{\text{res}}^{\text{PT}}, \mathcal{D}_{\text{res}}^{\text{SFT}}) = \mathbb{E}_{t \sim \mathcal{T}} \left[\text{Acc}(f_{\theta_{\text{SFT}}}(t)) \right], \quad (7.1)$$

where \mathcal{T} is a set of downstream reasoning tasks (math, science, code, general reasoning) and the expectation $\mathbb{E}_{t \sim \mathcal{T}}$ defines a single fine-tuned model that is evaluated across \mathcal{T} .

Our study can be summarized as optimizing the allocation of \mathcal{D}_{res} between pretraining and SFT:

$$\max_{\mathcal{D}_{\text{res}}^{\text{PT}}, \mathcal{D}_{\text{res}}^{\text{SFT}}} \mathcal{P}(\mathcal{D}_{\text{res}}^{\text{PT}}, \mathcal{D}_{\text{res}}^{\text{SFT}}) \quad \text{s.t.} \quad \mathcal{B} = |\mathcal{D}_{\text{res}}^{\text{PT}}| + |\mathcal{D}_{\text{res}}^{\text{SFT}}|, \quad (7.2)$$

where \mathcal{B} is the total budget of reasoning data available. This captures the trade-off of early, scale/diversity vs late, quality/complexity: $\mathcal{D}_{\text{res}}^{\text{PT}} \longleftrightarrow \mathcal{D}_{\text{res}}^{\text{SFT}}$

7.2.1 Model Architecture and Baseline

We select a hybrid transformer with a mixture of Mamba 2 (Dao & Gu, 2024), self-attention and FFN layers (NVIDIA, 2025a) with an 8B parameter for our base model, \mathcal{M} and pre-train from scratch for 1 trillion tokens. This size strikes a balance between computational feasibility and the capacity to learn complex reasoning patterns.

7.2.2 Data Pipeline

Our experimental design relies on a careful distinction between two categories of data: (1) a large-scale, general-purpose pretraining corpus, and (2) a reasoning-focused, instruction-tuning (SFT-style) datasets of varying quality and scale. This separation allows us to precisely control the injection of reasoning data at different stages of training.

General Pretraining Corpus ($\mathcal{D}_{\text{base}}$). For the base training corpus, we adopt the dataset introduced in NVIDIA (2025b), which contains 6.2T tokens drawn from high-quality Common Crawl, mathematics, and code sources. This corpus provides broad coverage of languages and technical domains, serving as the backbone of all pretraining experiments.

Reasoning Datasets (\mathcal{D}_{res}). To investigate the impact of data quality, diversity, and complexity, we curate four distinct reasoning-focused datasets in the question-answer format:

- **Large-Scale, Diverse Data (\mathcal{D}_{LDQ}).** To simulate a “quantity-over-quality” strategy, we employ the Nemotron-Pretraining-SFT-v1 dataset (NVIDIA, 2025b). This massive 268M samples of dataset offers extensive domain coverage, with a composition of approximately 56% math, 17% code, and 27% science and general-purpose reasoning. The dataset covers tasks ranging from simple Q&A to multi-turn dialogues, but with heterogeneous quality and reasoning depth, reflecting large-scale real-world availability.
- **Small-Scale, High-Quality Data (\mathcal{D}_{SHQ}).** To capture the effect of long chain-of-thought traces from strong teacher models, we include the dataset of Guha et al. (2025), comprising 1.2M carefully curated examples (71% math, 21% code, 8% science). Compared to \mathcal{D}_{LMQ} , this corpus is smaller, less diverse, but significantly higher quality, emphasizing detailed reasoning paths.
- **Large-Scale, Mixed-Quality Data (\mathcal{D}_{LMQ}).** To balance diversity with quality, we construct a combined dataset that is a straightforward union of the two datasets above: $\mathcal{D}_{\text{LMQ}} = \mathcal{D}_{\text{LDQ}} + \mathcal{D}_{\text{SHQ}}$, i.e., \mathcal{D}_{LMQ} contains 269.2M samples in total. This mix preserves large-scale coverage while injecting a fraction of curated, high-quality reasoning traces.
- **Answer-Length Filtered Data (\mathcal{D}_{ALF}).** To investigate the feature of data quality, we create a subset (7.1M samples) of \mathcal{D}_{LDQ} by retaining examples where the answer length exceeds 4096 tokens, based on the principle that longer responses often correspond to more complex CoT reasoning. This dataset allows us to isolate the impact of reasoning complexity in different training phases.

7.2.3 Synergy between Pretraining and SFT

In this work, we aim to disentangle the contribution of reasoning data when incorporated at different points in the training pipeline. We structure the study into three stages: (i)

large-scale **Pretraining**, where reasoning data may or may not be injected alongside the base corpus, (ii) **Supervised Finetuning (SFT)**, where pretrained models are further adapted on reasoning data of varying quality and diversity, and (iii) **Reinforcement Learning (RLVR)** to determine the sustainability of early reasoning gain in the final model.

Phase 1: Pretraining. Prior work has primarily explored reasoning supervision either on top of fully pretrained LLMs (Wang et al., 2025) or by introducing small amounts of long chain-of-thought (CoT) data into intermediate checkpoints (AI et al., 2025b). These approaches leave open two questions: how to inject reasoning data at scale during end-to-end pretraining, and whether the benefits persist when combined with high-quality base corpora. To address these questions, we pretrain all models *from scratch* for 1T tokens using a mix of $\mathcal{D}_{\text{base}}$ and different types of \mathcal{D}_{res} . Across all models, we keep the token ratio between $\mathcal{D}_{\text{base}}$ and \mathcal{D}_{res} fixed during pretraining. When a reasoning dataset is small, it is repeated so that the model still observes the same total volume of reasoning tokens. To correctly state the data distribution, we pretrain all models for 600B tokens using $\mathcal{D}_{\text{base}}$ followed by 400B tokens on a mixture of 80% $\mathcal{D}_{\text{base}}$ and 20% \mathcal{D}_{res} . This results in a constant budget of 80B reasoning tokens across all experiments. \mathcal{D}_{res} can be any one source of data among the three reasoning datasets: \mathcal{D}_{SHQ} , \mathcal{D}_{LDQ} , \mathcal{D}_{LMQ} . This token ratio has been maintained across all three pretraining runs with reasoning data.

Based on the reasoning data introduced, we train four distinct models:

- $\mathcal{M}_{\text{base}}$: This model serves as our **baseline** and is pretrained without any reasoning data.
- \mathcal{M}_{LDQ} : Pre-trained with large-scale, diverse \mathcal{D}_{LDQ} reasoning dataset along with $\mathcal{D}_{\text{base}}$.
- \mathcal{M}_{SHQ} : Pre-trained with \mathcal{D}_{SHQ} and $\mathcal{D}_{\text{base}}$ allowing us to isolate the effect of data quality versus the quantity and diversity of \mathcal{M}_{LDQ} .
- \mathcal{M}_{LMQ} : Finally, this model is exposed to our combined reasoning \mathcal{D}_{LMQ} dataset.

In the subsequent analysis, we use \mathcal{M}_{res} to denote the aggregate performance of the models pretrained with reasoning data, representing the average score across \mathcal{M}_{SHQ} , \mathcal{M}_{LDQ} , and \mathcal{M}_{LMQ} .

Phase 2: Supervised Finetuning. Following pretraining, each of the four model variants ($\mathcal{M}_{\text{base}}$, \mathcal{M}_{LDQ} , \mathcal{M}_{SHQ} , \mathcal{M}_{LMQ}) is adapted through supervised finetuning (SFT). This second phase is crucial for understanding the synergies, redundancies, and trade-offs between the data introduced during pretraining versus the SFT stage. To this end, we design a controlled set of SFT experiments, where each pretrained model is finetuned on different reasoning corpora introduced in Section 7.2.2 to address the following research questions:

- **The “Catch-Up” Hypothesis:** Can intensive SFT on high-quality reasoning data allow the baseline model, $\mathcal{M}_{\text{base}}$, to match or exceed the accuracy of models that were exposed to reasoning data during pretraining? This directly tests the criticality of early data injection versus late-stage specialization.
- **Impact of Pretraining Data Scale and Diversity:** We investigate how the scale and diversity of reasoning data used during pretraining influence the final model’s capacity to absorb high-quality instruction data. Specifically, we ask: *Does scaling up diverse reasoning data in pretraining provide lasting benefits even after all models are finetuned on the same high-quality SFT corpus?* By fine-tuning both the model pretrained on large, diverse data

(\mathcal{M}_{LDQ}) and on smaller, less diverse data (\mathcal{M}_{SHQ}) on the same high-quality SFT set, we can determine whether a broad or a deep initial exposure to reasoning yields a better foundation for downstream specialization.

- **Impact of SFT Data Quality and Complexity:** By fine-tuning all four base models on datasets of varying quality (\mathcal{D}_{LDQ} vs. \mathcal{D}_{SHQ}) and complexity (\mathcal{D}_{ALF}), we can measure the marginal utility of data quality at the SFT stage as a function of the model’s initial pretraining condition.

This design enables us to address three critical dimensions: (1) the **synergy** between pretraining and SFT data, (2) the **gains** of increasing SFT data scale when reasoning was already introduced in pretraining, and (3) the **role of data complexity and diversity** in determining whether reasoning supervision should be injected early, late, or across both stages. Together with the pretraining experiments, these SFT studies form a fully crossed setup, providing the first systematic assessment of how reasoning-style SFT data interacts with pretraining to shape the reasoning abilities of LLMs.

Phase 3: Reinforcement Learning. To further observe the impact of reasoning centric pretraining and heavy supervised finetuning, we deploy RL using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with verifiable rewards on top of the base models. Here we use NEMOTRON-CROSSTHINK (Akter et al., 2025) which has shown to be effective for multi-domain reasoning.

7.3 Experiments and Results

7.3.1 Training

Pretraining. To prepare base models, we pretrain a 8B LLM on our pretraining data blend till 1T tokens using 512 H100 80GB SXM5 GPUs. During training, we use the AdamW optimizer (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and weight decay of 0.1. We use a 8-way tensor and pipeline parallelism to train the model. We set the maximum value of learning rate to $3e^{-4}$, minimum to $3e^{-6}$, and use a batch size of 6M tokens with a 8192 context length.

Post-Training. After pretraining, each 8B LLM is finetuned on 4.8M reasoning samples from \mathcal{D}_{res} . SFT uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, warmup ratio 0.05, learning rate 5×10^{-6} , batch size 512, and context length 32k. We then apply GRPO via the veRL framework¹ for one epoch on NEMOTRON-CROSSTHINK data with constant LR 1×10^{-6} , batch size 128, PPO mini-batch 128, and context length 8192. Each step samples 128 prompts with 8 rollouts (temperature= 1.0, top- $p = 1.0$), and a KL penalty coefficient of 0.001.

7.3.2 Evaluation Metrics

We report average accuracies of all tasks under each of the following categories.

¹<https://github.com/volcengine/verl>

Base Model Evaluations. We conduct a thorough benchmark assessment to evaluate the generalizability of the base models, using a series of datasets using LM Eval Harness (Gao et al., 2024b).

- **General Purpose Reasoning (GPR_{PT} AVG).** We consider four standard commonsense and logical reasoning tasks in 0-shot: ARC challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), and reading comprehension task: RACE (Lai et al., 2017).
- **Math Reasoning (MATH_{PT} AVG).** We evaluate the math reasoning ability with two benchmarks—they encompass math challenges from elementary to college level complexity demanding qualitative reasoning (8-shot GSM8K (Cobbe et al., 2021b), 4-shot MATH-500 (Hendrycks et al., 2021e)).
- **Science Reasoning (SCIENCE_{PT} AVG).** We evaluate on 5-shot MMLU (Hendrycks et al., 2021b) and MMLU-Pro (Wang et al., 2024b) that spans multiple domains, from professional to academic, testing the model on specialized subjects.
- **Code Reasoning (CODE_{PT} AVG).** For code tasks (HumanEval (Chen et al., 2021a), MBPP (Odena et al., 2021)) we evaluate the EvalPlus variants along with the sanitization of generations (Liu et al., 2023c), in a 0-shot setup. We estimate avg@32, pass@1 from 32 generations per prompt.

SFT Model Evaluations. To evaluate the reasoning ability of different SFT models, we focus on reasoning centric benchmarks unlike in base model evaluations, where mostly focus on the generalizability of the LLM. We conduct evaluations using NeMo-Skills².

- **Math Reasoning (MATH_{SFT} AVG).** In addition to the GSM8K and MATH-500, we evaluate the models on two more complex math tasks—AIME24 and AIME25 (Veeraboina, 2023).
- **Science Reasoning (SCIENCE_{SFT} AVG).** On top of MMLU and MMLU-Pro, we evaluate on graduate level QA task: GPQA-Diamond (Rein et al., 2024).
- **Code Reasoning (CODE_{SFT} AVG).** We choose LiveCodeBench (Jain et al., 2025) to test complex code reasoning ability.
- **Instruction Following (INS_{SFT} AVG).** For broader evaluation on diverse capabilities, we use IFEval (Zeng et al., 2024).

We report Pass@1 average of 16 runs for AIME-2024, AIME-2025 and average of 4 runs for MATH-500, GSM8K, MMLU, MMLU-Pro, GPQA-Diamond, LiveCodeBench and IFEval.

RL Model Evaluations. In this phase, we evaluate the models on complex reasoning tasks such as AIME24,25, MATH-500, GSM8K, MMLU, MMLU-Pro, GPQA-Diamond, LiveCodeBench following the evaluation metric in SFT phase.

7.3.3 Results

Immediate Foundational Gains from Reasoning Data in Pretraining. Table 7.1 shows the average accuracies of our four model variants immediately after the 1T token pretraining phase (see Table 13.1 for individual benchmarks). The results provide clear evidence

²<https://github.com/NVIDIA/NeMo-Skills>

that integrating reasoning-style corpora from the start builds a significantly more capable foundation. Every model exposed to reasoning data surpasses $\mathcal{M}_{\text{base}}$. \mathcal{M}_{res} , average of the

Model	Average	MATH _{PT} AVG	SCIENCE _{PT} AVG	CODE _{PT} AVG	GPR _{PT} AVG
$\mathcal{M}_{\text{base}}$	52.70	47.17	47.13	40.89	75.63
\mathcal{M}_{SHQ}	54.98	52.60	46.90	44.32	76.09
\mathcal{M}_{LDQ}	64.09	75.56	54.38	49.94	76.48
\mathcal{M}_{LMQ}	64.07	72.37	54.49	52.60	76.83
\mathcal{M}_{res}	61.05	66.84	51.92	48.95	76.46

Table 7.1: **Average Accuracies of base models trained without or with varying \mathcal{D}_{res} .** Pretraining with diverse reasoning data yields immediate gains, with scale and diversity driving math and code improvements, more than quality. \mathcal{M}_{res} represents the average of \mathcal{M}_{SHQ} , \mathcal{M}_{LDQ} , and \mathcal{M}_{LMQ} .

three reasoning-augmented variants trained under the same 1T token budget, improves over $\mathcal{M}_{\text{base}}$ by +8.35% on overall average accuracy. The largest improvements come from models trained on large-scale, diverse data; \mathcal{M}_{LDQ} achieves highest average, driven by a +28.4% gain in mathematics and a +9% gain in code over the baseline. Interestingly, the smaller, less diverse, high-quality dataset (\mathcal{M}_{SHQ}) provides a modest lift, suggesting that at this early stage, the scale and diversity of the reasoning data are more critical than its curated quality for establishing a broad and robust reasoning foundation. Our experiments with a 1.2B Transformer (see Table 13.6) demonstrate that this front-loading strategy yields consistent, scalable performance gains—confirming the robustness of our approach across varying architectures.

Pretraining Advantage is Maintained and Amplified Post-SFT. We evaluate whether a strong SFT phase can close the accuracy gap established during pretraining with diverse reasoning data \mathcal{D}_{res} . At the same time, we examine whether the inclusion of such data causes the model to overfit and reduce generalization, thereby diminishing subsequent post-training gains. We finetune each pretrained model on three reasoning datasets (\mathcal{D}_{SHQ} , \mathcal{D}_{LDQ} , \mathcal{D}_{LMQ}), producing 12 models in total. We report the average results in Table 7.2 and include the full breakdown in Table 13.5.

Model	Average	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{base}} + \text{SFT}$	26.62	34.48	20.92	7.09	43.98
$\mathcal{M}_{\text{res}} + \text{SFT}$	35.92	40.61	34.77	16.75	51.52

Table 7.2: **Average Accuracies of SFT models pretrained with varying \mathcal{D}_{res} .** SFT amplifies the pretraining advantage—models with reasoning-rich pretraining significantly outperform baseline.

The results in Table 7.2 indicate that the advantage gained during the pre-training phase not only persists but is amplified. The group of models pretrained with reasoning data ($\mathcal{M}_{\text{res}} + \text{SFT}$) outperforms the baseline group ($\mathcal{M}_{\text{base}} + \text{SFT}$) by a significant 9.3% on average. This result strongly refutes the “catch-up” hypothesis, showing that SFT is not a substitute for a strong reasoning foundation built during pretraining. While recent works have found reasoning-centric post-training to be most effective on math domains, the improvement on

science is minimal (Prabhakar et al., 2025; Luo et al., 2025a; Huan et al., 2025). However, the accuracy disparity in our findings is most prominent in science domains, an area often overlooked in reasoning-focused post-training work. This suggests that pretraining with reasoning data does more than teach facts; it helps the model develop effective internal representations for abstract and logical structures to enhance problem solving ability across domains. It does not overfit the model rather infuses the critical thinking ability that comes into full potential after post-training (Appendix 13.2). Consequently, the model’s capacity to absorb and leverage the SFT data is fundamentally enhanced, leading to greater learning efficiency and a higher performance ceiling. SFT acts as a powerful enhancer, but its ultimate effectiveness is constrained by the quality of the foundation established during pretraining.

Model	Avg.	Math Reasoning				Science & Code Reasoning			
		MATH-500	GSM8K	AIME24	AIME25	GPQA	MMLU	MMLU-PRO	LCB
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}} + \text{RL}$	37.92	72.05	83.83	12.29	16.04	28.16	41.10	36.69	13.16
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}} + \text{RL}$	56.66	87.13	93.07	45.21	33.96	31.69	72.91	56.91	32.43

Table 7.3: Average accuracies of RL models pretrained and fine-tuned with varying \mathcal{D}_{res} . Introducing reasoning data early provides significant reasoning boost after post-training.

Pretraining Strategy Dictates Final Accuracy on Expert-Level Tasks. The final RL phase reveals the definitive impact of our pretraining interventions, particularly on expert-level reasoning benchmarks. We select $\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}}$ and $\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}}$ finetuned using \mathcal{D}_{SHQ} as our two extreme pretraining backbones. As shown in Table 7.3, the accuracy gap between the two models continues to diverge, with the fully-aligned \mathcal{M}_{LMQ} models achieving a 18.57% lead over the $\mathcal{M}_{\text{base}}$ model on average. The most striking results appear on the highly challenging AIME competition math problems, where the reasoning-pretrained models deliver a 39.32% improvement over the baseline. This provides conclusive evidence that early investment in reasoning data yields compounding returns, becoming the decisive factor in achieving frontier accuracies on the most demanding tasks.

7.4 Ablations

Does the scale and diversity of the reasoning data matter in Pretraining? As detailed in Table 7.1, plainly increasing size and diversity of \mathcal{D}_{res} in pretraining has significant improvement on the base model. The model pretrained on large, diverse data (\mathcal{M}_{LDQ}) achieves an absolute +9.09% average gain over the model trained on the smaller, less diverse corpus (\mathcal{M}_{SHQ}), with the largest gains observed in math, science, and code—domains that explicitly demand structured reasoning. GPR_{PT} AVG shows limited sensitivity to diversity due to the nature of tasks that require commonsense and general knowledge. In contrast, scaling \mathcal{D}_{LDQ} with \mathcal{D}_{SHQ} (high-quality but less diverse) as in \mathcal{M}_{LMQ} provides minimal further benefit on the reasoning tasks—underscoring that broad exposure to diverse reasoning patterns during pretraining is impactful for building a strong foundation.

The Pretraining Advantage Persists and Resists “Catch-Up” Attempts via SFT. A central question is whether a model without a reasoning-rich pretraining ($\mathcal{M}_{\text{base}}$) can compensate for this deficit by undergoing a more intensive SFT phase. We test this “catch-up” hypothesis by fine-tuning $\mathcal{M}_{\text{base}}$ with two times more epochs using the same SFT data (\mathcal{D}_{SHQ}). The results in Table 7.4 prove this hypothesis false. While doubling the SFT epochs improves the

Model	Average	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}}$	29.92	42.79	35.83	10.48	30.59
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}}(2\times\text{epochs})$	34.01	48.05	40.69	14.60	32.70
$\mathcal{M}_{\text{SHQ}} + \text{SFT}_{\text{SHQ}}$	37.33	50.52	40.00	24.76	34.06
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{SHQ}}$	46.70	60.79	50.67	28.57	46.79
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}}$	50.95	64.67	53.74	35.55	49.82

Table 7.4: **Impact of diverse pretraining \mathcal{D}_{res} on SFT phase.** Doubling SFT for the baseline fails to “catch up” to reasoning-pretrained models, while the latent advantage of the mixed-quality pretraining (\mathcal{M}_{LMQ}) emerges, making it the top performer.

baseline’s average score by 4.09%, this enhanced baseline **still fails to match** the performance of even our weakest reasoning-pretrained model, $\mathcal{M}_{\text{SHQ}} + \text{SFT}_{\text{SHQ}}$ (+3.32%). This provides strong evidence that pretraining instills a foundational reasoning capability that cannot be fully replicated by simply scaling the SFT phase.

Post-SFT, high-quality data reveals latent value. The downstream consequences of these pretraining choices become more nuanced after SFT. To isolate and test whether these effects persist into post-training, we finetune all base models with the same high-quality SFT recipe (\mathcal{D}_{SHQ}). Results in Table 7.4 confirm that models pretrained on diverse corpora continue to substantially outperform less diverse counterparts even after SFT, confirming that a diverse pretraining foundation enhances a model’s capacity to benefit from SFT. More surprisingly, while the immediate gains of scaling with high-quality but narrow data (\mathcal{M}_{LMQ}) were muted at the pretraining stage, SFT reveals a latent advantage: \mathcal{M}_{LMQ} achieves an additional +4.25% gain over \mathcal{M}_{LDQ} post-SFT. This reveals a critical finding that high-quality but less diverse data may act as a *complementary amplifier*, whose benefits emerge after alignment—underlining the latent impact of quality of data during the pretraining.

Model	Average	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}}$	29.92	42.79	35.83	10.48	30.59
$\mathcal{M}_{\text{res}} + \text{SFT}_{\text{LMQ}}$	31.21	30.91	27.73	9.79	56.41
$\mathcal{M}_{\text{res}} + \text{SFT}_{\text{LDQ}}$	31.54	32.28	28.43	10.85	54.61
$\mathcal{M}_{\text{res}} + \text{SFT}_{\text{SHQ}}$	44.99	58.66	48.14	29.63	43.56

Table 7.5: **Impact of diverse SFT \mathcal{D}_{res} on SFT phase.** Fine-tuning on the small, high-quality corpus (\mathcal{D}_{SHQ}) is highly effective, while using large, diverse corpora (\mathcal{D}_{LDQ}) degrades reasoning.

SFT is dominated by data quality, not diversity. We finetune all reasoning-pretrained models (\mathcal{M}_{res}) on each of our distinct reasoning datasets, and report the averaged results in Table 7.5. The findings reveal a striking contrast: while diversity is beneficial in pretraining, blindly scaling diverse reasoning data during SFT degrades performance. Models trained with \mathcal{D}_{LDQ} or \mathcal{D}_{LMQ} during SFT underperform relative to those finetuned on the smaller, high-quality, long-CoT dataset, \mathcal{D}_{SHQ} , despite having been exposed to reasoning data during pretraining. In fact, the use of large-scale, mixed-quality data at the SFT stage not only erodes the benefits of reasoning-rich pretraining but can even lead to worse outcomes than

Model	$\mathcal{D}_{\text{base}} : \mathcal{D}_{\text{res}}$	Overall	MATH _{PT} AVG	SCIENCE _{PT} AVG	CODE _{PT} AVG	GPR _{PT} AVG
\mathcal{M}_{LMQ}	80 / 20	64.07	72.37	54.49	52.60	76.83
	90 / 10	63.97	75.24	53.21	50.19	77.23
	60 / 40	67.28	79.63	55.73	56.47	77.31

Table 7.6: **Effect of varying reasoning data ratio during pretraining.** Increasing the reasoning proportion improves reasoning-focused benchmarks while preserving general-domain performance.

the baseline $\mathcal{M}_{\text{base}}$ finetuned with \mathcal{D}_{SHQ} in math, code, and science tasks which benefit from reasoning. This result confirms the widely held view that data quality and long reasoning data is critical for effective SFT (Zhou et al., 2023; Zhao et al., 2024; Prabhakar et al., 2025). Our findings, however, extend this understanding by showing that simply applying high-quality data at every stage is not optimal. Instead, the most effective strategy is **asymmetric**: pretraining benefits most from broad and diverse reasoning data to establish generalizable priors, whereas SFT requires high-quality, reasoning-heavy data for targeted refinement.

Reasoning Ratio Sensitivity and Its Interaction with Alignment. As outlined in Section 7.2.3, the ratio between reasoning data and general pretraining data is fixed at 20% on 400B tokens in our main experiments to ensure fair comparison across \mathcal{D}_{SHQ} , \mathcal{D}_{LDQ} , and \mathcal{D}_{LMQ} . While this provides a controlled setting, the optimal ratio is inherently empirical and may vary across domains and datasets. To study the sensitivity of this hyperparameter, we conduct additional experiments by varying the reasoning proportion using $\mathcal{D}_{\text{res}} = \mathcal{D}_{\text{LMQ}}$.

As detailed in Table 7.6, increasing the reasoning ratio from 20% to 40% improves overall pretraining accuracy and substantially boosts math, science, and code benchmarks. Reducing it to 10% produces a mild drop in overall performance. These results indicate that the reasoning signal scales positively with its proportion during pretraining.

Model	Overall	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}}$	50.95	64.67	53.74	35.55	49.82
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}} [60/40]$	52.63	67.71	54.19	43.81	44.81

Table 7.7: **Effect of pretraining reasoning ratio on downstream SFT performance.** Higher reasoning proportion improves reasoning benchmarks but slightly reduces instruction-following metrics.

We further examine how these ratios affect downstream performance after SFT. Using \mathcal{D}_{SHQ} for fine-tuning under the same SFT recipe, we compare the 80/20 and 60/40 pretrained models in Table 7.7. The 60/40 model continues to improve reasoning benchmarks after SFT, particularly in math and code, but shows a decline in instruction-following performance. This aligns with the breadth–alignment trade-off discussed in Table 13.9. Increasing reasoning proportion strengthens structured reasoning capabilities, while general pretraining data contributes broader stylistic and formatting exposure that supports instruction-following flexibility. In summary, these experiments indicate that the reasoning ratio is a meaningful control knob. Higher proportions consistently strengthen reasoning performance both

Model	Average	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{LDQ} + \text{SFT}_{LDQ}$	32.84	28.38	35.22	10.16	57.61
$\mathcal{M}_{LDQ} + \text{SFT}_{2 \times LDQ}$	32.99	23.46	39.65	11.75	57.10
$\mathcal{M}_{LDQ} + \text{SFT}_{ALF}$	42.66	60.95	47.29	22.54	39.87
$\mathcal{M}_{LDQ} + \text{SFT}_{ALF'}$	43.04	61.61	45.78	22.53	42.23

Table 7.8: **Impact of scaling reasoning data in SFT phase.** Naively doubling mixed-quality data is detrimental to math reasoning, whereas targeted scaling of high-quality data yields consistent gains.

before and after SFT, but may modestly reduce alignment-sensitive metrics. The optimal balance may therefore depend on the target deployment domain, and systematic exploration across datasets remains an important direction for future work.

How should we expand reasoning data during SFT? We next ablate the effect of scaling reasoning data during the SFT phase by contrasting two strategies: (i) scaling with data of *similar quality and diversity*, and (ii) scaling with data of *higher quality and reasoning depth*. As shown in Table 7.8, simply doubling the amount of diverse but mixed-quality data yields negligible improvement in average accuracy with a 4.92% drop in math accuracy—suggesting that increasing the volume of noisy or shallow reasoning data may dilute the useful signal and actively harm reasoning-specific domains. The small gains in science and code do not offset this regression, highlighting the limits of quantity-driven scaling in SFT. In contrast, when scaling \mathcal{D}_{ALF} with high-quality \mathcal{D}_{SHQ} (\mathcal{D}'_{ALF}), the average accuracy improves further, with math and instruction-following tasks benefiting most. Importantly, this qualitative expansion is achieved with only a marginal increase in dataset size (0.4% more samples). These contrasting outcomes provide clear evidence that SFT is a phase of targeted refinement, not broad data absorption; the most effective scaling strategy is to strategically enhance the training corpus with high-quality, reasoning-intensive examples.

7.5 Related Work

Reasoning in Pretraining and Midtraining. Cheng et al. (2024) study *instruction pretraining* by converting raw text into short QA pairs and report gains on general-purpose reasoning tasks that require minimal reasoning. While effective for broad linguistic alignment, their setup does not explicitly target reasoning-intensive domains such as mathematics, graduate level science, or code. Moreover, their pipeline of self-distilled instruction generation demonstrates that Instruct-PT outperforms vanilla PT after instruction tuning, but it does not assess whether these marginal pretraining gains persist once models undergo reasoning-heavy SFT and reinforcement learning. In contrast, our work systematically varies the *complexity, quantity, and diversity* of reasoning-style SFT data—containing intermediate thoughts and answers—across both pretraining and SFT, allowing us to probe whether early exposure yields durable downstream advantages.

More recent efforts have begun to explore the interplay between pretraining and instruction tuning. Liang et al. (2025) augment the instruction-tuning pool to better align with the distribution of pretraining data, reinforcing consistency between the two stages. While complementary in spirit, their method is applied only during SFT and does not address

whether reasoning-specific supervision at the pretraining stage provides sustained benefits. Similarly, Wang et al. (2025); AI et al. (2025b) introduce a *mid-training* phase, continuing pretraining on a small but high-quality reasoning dataset before SFT and RLVR. They report substantial downstream gains, particularly in mathematics benchmarks, highlighting the promise of mid-training interventions. However, because their corpus is heavily math-centric, it is difficult to disentangle whether the improvements stem from scale, complexity, or domain diversity, and the generalizability to science or code remains unclear.

A complementary direction is pursued by Gandhi et al. (2025b), who inject algorithmically generated “cognitive behavioral” reasoning traces during mid-training, demonstrating improvements after reinforcement learning. This underscores the potential of early reasoning supervision but remains limited in scope: the interventions are restricted to small datasets and narrow tasks, leaving open questions about scalability, diversity, and phase-specific allocation of reasoning data. Our work builds on these insights by conducting the first systematic, large-scale analysis of reasoning data across both pretraining and SFT, providing a principled framework for understanding *when* and *how* reasoning supervision should be applied.

How next-token prediction can support reasoning? Recent studies indicate that large language models acquire non-trivial reasoning skills during the pretraining phase itself, even before any reinforcement learning (RL) or supervised fine-tuning (SFT) is applied (AI et al., 2025b). This finding challenges the common assumption that complex reasoning (e.g. multi-step problem solving with self-correction) only arises from post-training alignment or fine-tuning. Instead, evidence shows that standard next-token pretraining on vast data can already instill mechanisms analogous to those seen in RL- or SFT-tuned models. A natural question is how a *simple* objective—predicting the next token—can give rise to reasoning. Several lines of work provide concrete evidence.

Autoregressive next-token prediction is equivalent to modeling the full sequence distribution via the chain rule of probability: $P(x_0, \dots, x_T) = \prod_t P(x_t | x_{<t})$. Minimizing cross-entropy over this factorization therefore trains the model to approximate the true conditional $P(x_t | x_{<t})$ at each step. Whenever the data distribution contains long-range dependencies (e.g., conclusions that depend on earlier premises, or answers that depend on multi-step derivations), the Bayes-optimal predictor must capture those dependencies to minimize loss. In other words, if the training corpus includes text that reflects logical or causal structure, the next-token objective *necessarily* rewards parameters that encode that structure; there is no statistical shortcut that avoids it. This does not guarantee that finite models learn it, but it shows that the objective is not inherently hostile to reasoning—rather, it is aligned with modeling any conditional dependency present in the data.

Butkus & Kriegeskorte (2025) provide direct empirical evidence that GPT-style transformers trained *only* on next-token prediction can discover structural causal models (SCMs) and answer counterfactual queries. In controlled settings, the network generalizes to counterfactual questions about SCMs where it has seen only interventional data, suggesting it composes discovered causal structure with a learned counterfactual inference algorithm. The implicit “mental” SCM can be decoded from the model’s residual stream and manipulated with predictable effects on outputs. This challenges the claim that statistical prediction cannot support causal reasoning: the data (e.g., natural language describing interventions

and causes) carries the necessary information, and next-token prediction is sufficient to drive the emergence of internal causal models.

Wang et al. (2024a) show that transformers can learn *implicit* reasoning (e.g., composition and comparison over parametric knowledge) exclusively through next-token prediction, but only after *grokking*—extended training far beyond the point of overfitting. During grokking, generalizing circuits form and gradually become more efficient than memorizing circuits; the formation of these circuits mirrors the development of specialized attention patterns (e.g., induction heads) that support in-context learning. For challenging reasoning tasks with large search spaces, fully grokked transformers achieve near-perfect accuracy using parametric memory, outperforming retrieval-augmented frontier models. This provides an algorithmic explanation: next-token prediction does not explicitly reward “reasoning,” but minimizing loss over reasoning-heavy data eventually favors parameters that implement efficient, generalizing algorithms over brittle memorization.

Riechers et al. (2025) argue that in-context learning (ICL) arises as a *mathematically predictable* consequence of next-token pretraining rather than an exotic emergent quirk. Their information-theoretic framework shows that models trained on token sequences, especially from non-ergodic sources, necessarily adapt to context in a way that reduces in-distribution loss—i.e., they exhibit ICL dynamics. Experiments reproduce phase transitions (e.g., induction-head formation) and power-law scaling. Since in-context adaptation is a form of using the prompt to “reason” about the task, this provides a theoretical link: next-token pretraining implies context-dependent prediction, which in turn supports reasoning-like behavior at inference time.

In sum, simple next-token prediction can support reasoning because (i) the chain rule and cross-entropy minimization align the objective with any conditional structure in the data, including logical and causal dependencies; (ii) empirical work shows that NTP-trained models can learn causal structure and counterfactual inference (Butkus & Kriegeskorte, 2025); (iii) grokking yields generalizing circuits that implement implicit reasoning (Wang et al., 2024a); and (iv) next-token pretraining provably implies in-context learning, which underpins reasoning from context (Riechers et al., 2025). These results do not imply that NTP is *sufficient* for all forms of reasoning at scale, but they provide logical, statistical, and algorithmic evidence that it is a viable foundation.

7.6 Discussion: Mechanistic signatures of front-loading reasoning data

To understand why front-loading reasoning data improves downstream reasoning behavior, we compare the internal representations of **FLR** and **Base** on the same short reasoning prompts. Rather than treating the gains of FLR as a purely empirical phenomenon, these analyses help reveal *where* in the network the two models begin to differ, and what kind of computation FLR appears to encourage. Concretely, for each layer we measure: (1) **Centered Kernel Alignment (CKA)** between FLR and Base hidden states, where higher values indicate more similar representations; (2) the **mean L2 difference** between hidden states, which quantifies how far activations move; (3) **logit similarity** across generation positions; and (4) **token-level prediction entropy**, which indicates how uncertain the model is while generating reasoning traces.

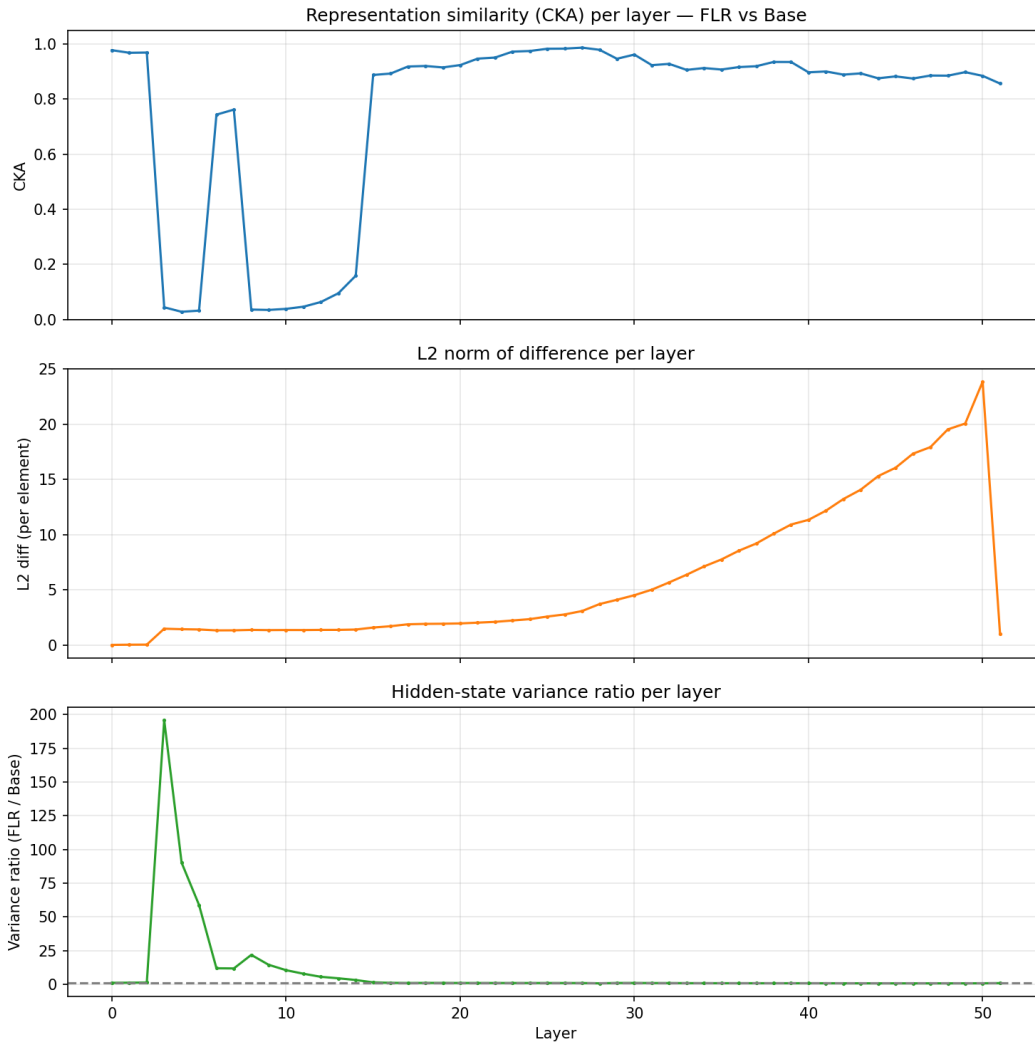


Figure 7.2: Layerwise comparison between FLR and Base. Top: CKA similarity per layer. Middle: mean hidden-state L2 difference. Bottom: hidden-state variance ratio.

The most important pattern is that FLR does not differ from Base uniformly across the network. As shown in [Figure 7.2](#), the earliest layers remain relatively similar, suggesting that both models retain a common substrate for lexical and syntactic processing. The strongest divergence emerges in a concentrated band of early-to-middle layers, where CKA drops sharply and hidden-state variance increases substantially. This implies that reasoning-heavy pretraining does not simply perturb the network globally; instead, it selectively rewrites the part of the model that transforms surface token sequences into structured intermediate states. A useful interpretation is that FLR changes the stage at which the model begins to organize a problem into latent substeps, constraints, and partial solutions. Thus, the benefit of FLR is not merely that it exposes the model to more reasoning traces, but that it teaches the model to enter a reasoning-oriented mode of computation much earlier in the forward pass.

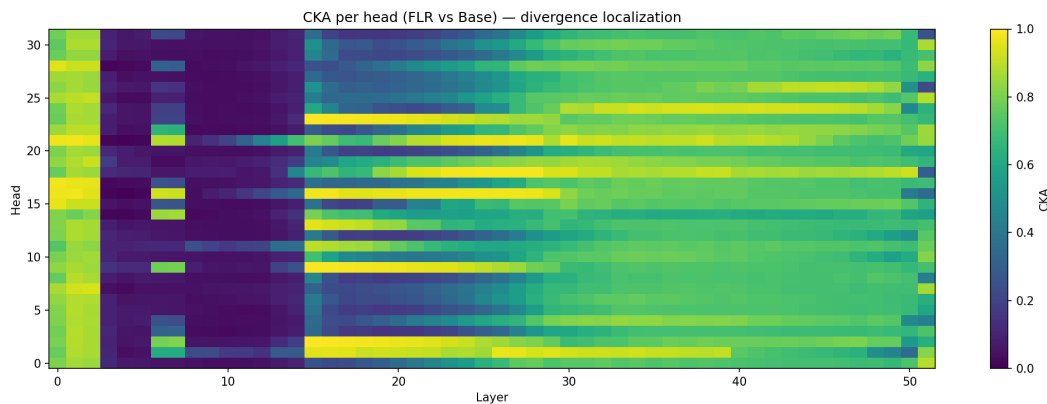


Figure 7.3: Attention-head level CKA between FLR and Base across layers. Lower similarity indicates stronger head-level divergence.

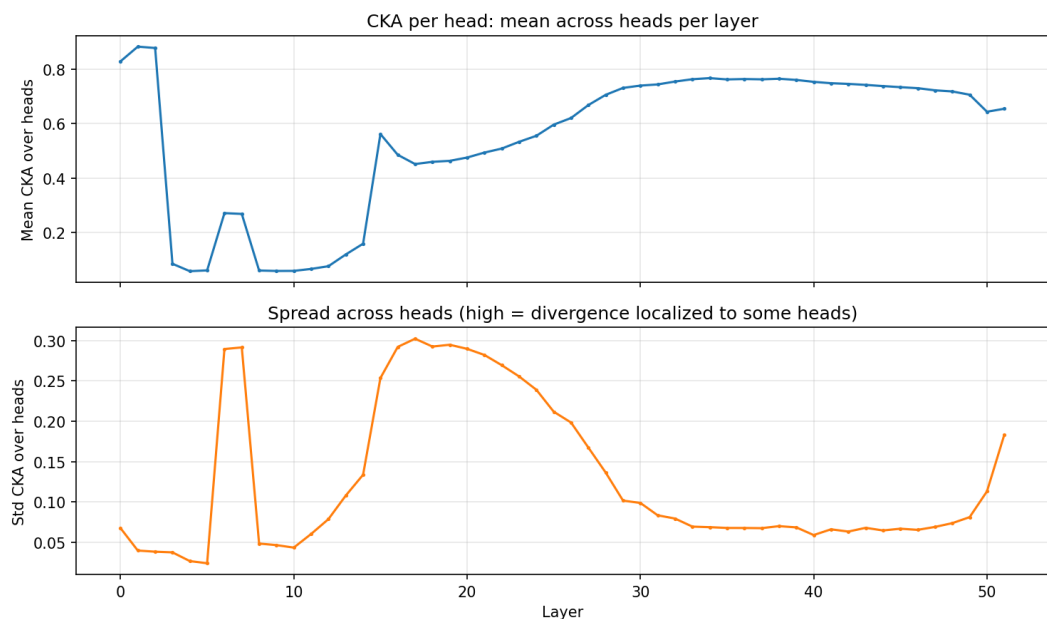


Figure 7.4: Summary statistics of attention-head divergence between FLR and Base. Top: mean CKA across heads per layer. Bottom: spread across heads, where higher spread indicates that divergence is concentrated in a subset of heads.

The variance and norm analysis further supports this view. In the same layers where FLR diverges most strongly from Base, the hidden-state variance ratio spikes dramatically (Figure 7.2), while the layerwise norm and variance trajectories in Figure 7.5 remain broadly stable rather than pathological. This suggests that FLR expands the representational space available to the model without destabilizing the overall computation. In effect, the model appears to learn a richer set of latent directions that can separate different reasoning states from one another. A base pretrained model can still produce step-by-step text when prompted, but its hidden states seem less specialized for carrying intermediate reasoning variables. FLR, by contrast, appears to allocate more capacity to the internal bookkeeping required for

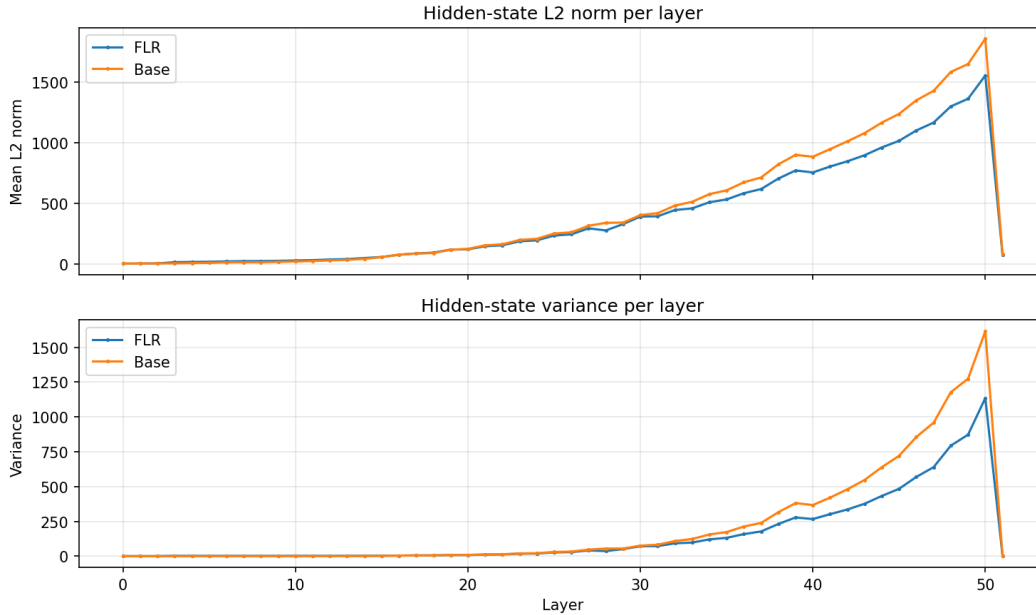


Figure 7.5: Hidden-state norm and variance per layer for FLR and Base.

multi-step inference. This is precisely the kind of inductive bias one would expect to help mathematical, logical, and procedural reasoning tasks.

The head-level results reveal that this change is also selective at the circuit level. As seen in [Figure 7.3](#) and [Figure 7.4](#), divergence is not evenly distributed across all attention heads. Some heads remain close to Base, while others change substantially, especially in the early and middle layers. This pattern suggests functional specialization rather than wholesale disruption. FLR does not need to rewrite the entire network to improve reasoning; it only needs to modify the heads that support operations such as long-range dependency tracking, decomposition into subproblems, and maintenance of intermediate state. This selective reconfiguration is a plausible explanation for why front-loading reasoning can improve reasoning performance without sacrificing the broad language competence inherited from general web-scale pretraining. The general linguistic machinery remains largely intact, while a subset of heads becomes more dedicated to reasoning-relevant computation.

These internal changes also translate into meaningful differences in the output distribution. As shown in [Figure 7.6](#), the two models become progressively less aligned in their logits as generation proceeds: logit correlation decreases, while KL and Jensen–Shannon divergences remain nontrivial across positions. This indicates that the representational changes introduced by FLR are behaviorally consequential. The models may begin from similar prompt-conditioned states, but once a multi-step solution unfolds, FLR increasingly favors different continuations than Base. This is exactly the regime where reasoning quality matters most. Strong reasoning performance depends on maintaining a coherent trajectory over many tokens, not simply predicting the first response token correctly. The logit divergence therefore suggests that FLR is not only learning a different internal representation, but also using that representation to sustain a different generation policy over the course of a reasoning trace.

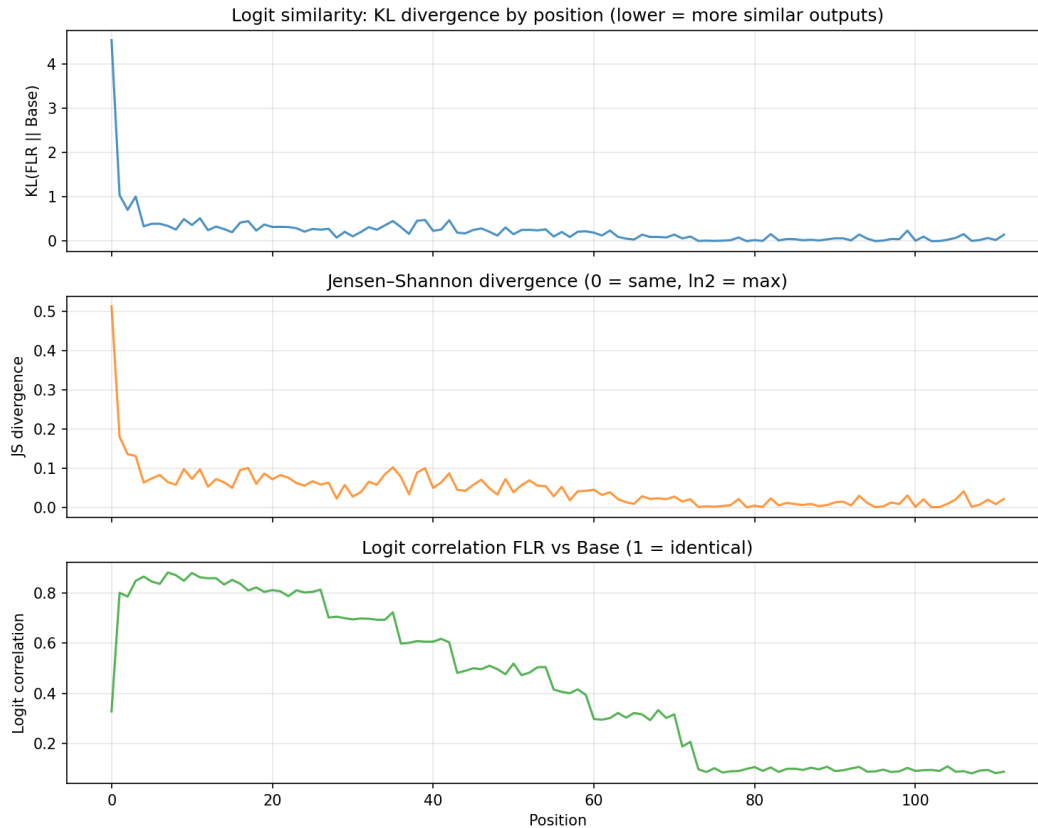


Figure 7.6: Output-space comparison between FLR and Base across generation positions. Top: KL divergence. Middle: Jensen-Shannon divergence. Bottom: logit correlation.

The entropy analysis provides an additional clue about how FLR helps. In [Figure 7.7](#), FLR is often slightly lower-entropy than Base, especially earlier in generation, while the gap narrows later. This pattern is more informative than a simple claim that FLR is “more confident.” Instead, it suggests that FLR helps the model commit earlier to a structured reasoning path. A broad-domain pretrained model often treats a reasoning prompt as one continuation style among many plausible textual modes. In contrast, FLR has repeatedly seen chain-of-thought and step-by-step solutions during pretraining, so it can enter that mode more decisively. Once the reasoning trajectory has been established, the entropy of the two models becomes more similar. This indicates that the main gain of FLR is not blanket confidence, but better initialization and stabilization of the reasoning process. That is especially valuable because many reasoning failures begin with an unstable or poorly chosen early decomposition, after which the rest of the sequence simply follows the wrong path.

Overall, these results suggest that front-loading reasoning data improves reasoning by reshaping the model’s internal computation before post-training begins. FLR appears to build internal circuits that treat reasoning as a native computational regime rather than a superficial response format learned only during later instruction tuning. The largest changes occur not at the input or output ends alone, but in the early-to-middle computational layers where the model likely forms intermediate abstractions and routing patterns. This provides

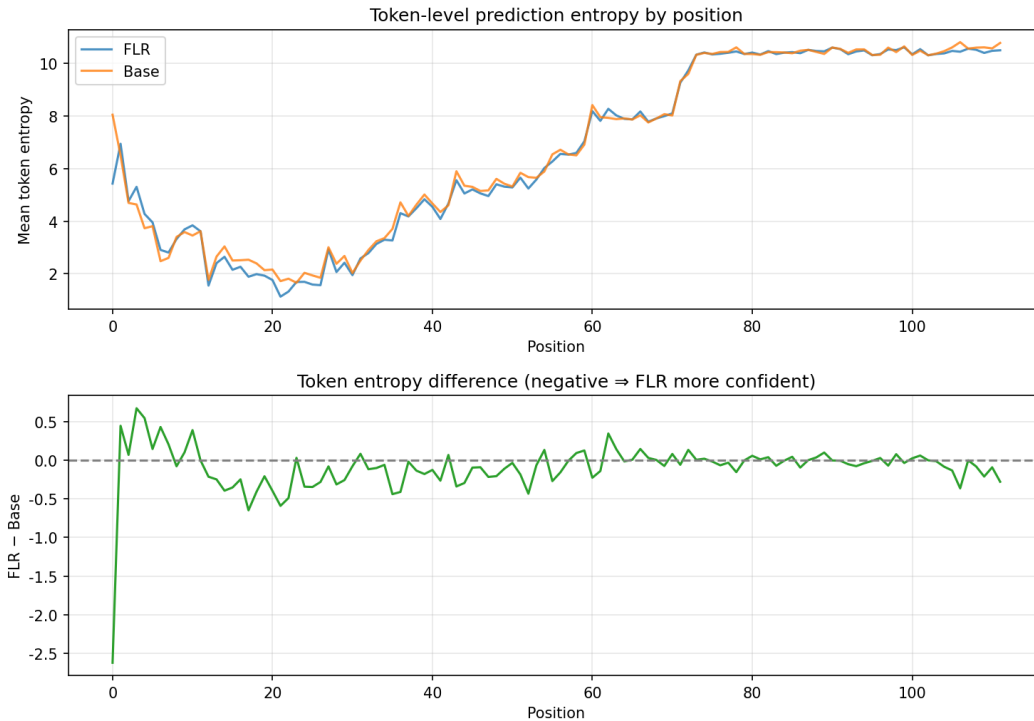


Figure 7.7: Token-level prediction entropy by generation position for FLR and Base, and their difference. Negative values indicate that FLR is more confident.

a mechanistic explanation for why FLR can yield stronger downstream reasoning: post-training is no longer trying to induce reasoning behavior on top of a purely general-purpose language model, but is instead refining a backbone that already contains the latent structure needed for decomposition, state tracking, and multi-step inference.

7.7 Summary

Our study provides the first systematic investigation of how reasoning data, varying in scale, diversity, and quality, influences LLMs across the entire training pipeline. We show that reasoning must be introduced early: front-loading into pretraining creates durable foundations that post-training alone cannot recover. Crucially, we uncover an asymmetric allocation principle—diversity drives pretraining effectiveness, while quality governs SFT—providing a clear, actionable blueprint for data strategy. Further, we demonstrate that high-quality pretraining data can yield latent benefits activated only during SFT, and that naive SFT scaling with noisy data can be actively harmful. Collectively, these findings challenge the conventional division between pretraining and reasoning, positioning reasoning-aware pretraining as a critical ingredient in building more capable, generalizable, and compute-efficient language models.

Part III

Redesign the Learning Pipeline to Enable Reasoning Emergence

Chapter 8

Bridging Gap between Pretraining and Post-training

Recent advances in large-scale models have shown that scaling data, parameters, and compute can produce systems with strong pattern recognition abilities. However, many models still struggle with structured reasoning unless they undergo extensive post-training procedures such as supervised fine-tuning or reinforcement learning with curated feedback. These approaches are often applied after pretraining, treating reasoning as a capability that is added later rather than developed during the core learning phase. As a result, improvements obtained during post-training can be fragile and may not generalize well beyond the specific tasks or distributions used for alignment.

This observation raises an important question: can the training pipeline itself be designed to encourage reasoning earlier in the learning process? If models begin to acquire reasoning-friendly representations during pretraining, later stages of adaptation may build upon a stronger foundation rather than attempting to retrofit reasoning onto representations that were optimized primarily for surface-level prediction. Achieving this requires rethinking how models learn from raw data and what types of signals guide representation learning at scale.

One direction toward this goal lies in reconsidering how self-supervised objectives expose information to the model. Many modern pretraining frameworks rely on reconstructing partially hidden inputs to learn contextual representations. While effective, these objectives often hide portions of the data at random, implicitly assuming that all parts of the input contribute equally to learning. In practice, however, meaningful understanding often depends on identifying the elements that distinguish one concept, event, or relation from another. Designing training signals that emphasize such informative differences can encourage models to capture the semantic structure required for complex reasoning tasks.

A complementary challenge concerns how models generate predictions during training. Standard next-token prediction encourages models to directly map context to the next output token, leaving little incentive for forming intermediate reasoning steps that might improve those predictions. Human comprehension, in contrast, frequently involves internal deliberation before producing an answer. Incorporating mechanisms that reward useful

intermediate reasoning during pretraining may therefore help models organize knowledge in ways that support long-range inference and better integration of contextual information.

This chapter explores how modifying early training objectives along these two dimensions can help shape the emergence of reasoning abilities. By guiding models to focus on informative signals during representation learning and by encouraging internal reasoning processes during prediction, the training pipeline can provide stronger foundations for downstream capabilities. Rather than relying solely on post-training to induce reasoning, these approaches aim to influence learning from the outset so that reasoning emerges as a natural consequence of pretraining.

8.1 Problem Statements

8.1.1 Difference Masking in Pretraining

Large pretrained models are typically trained on broad, heterogeneous corpora that span many domains and topics. While this diversity enables strong general-purpose representations, it also introduces a gap when adapting models to specialized domains. A common strategy to address this challenge is *continued pretraining*, where a pretrained model is further trained on unlabeled data drawn from a target domain before downstream task fine-tuning (Gururangan et al., 2020; Dery et al., 2023). In most cases, this adaptation uses the same self-supervised objective employed during initial pretraining, such as masked prediction, where portions of the input are hidden and the model learns to reconstruct them from context.

Despite its effectiveness, this approach typically assumes that the choice of which tokens to mask is inconsequential, and masking locations are often selected randomly. However, when adapting to a new domain, not all parts of the input carry equal importance for representation learning. A pretrained model already captures broad linguistic patterns from its original training distribution, but may lack strong representations for concepts that are distinctive to the new domain. Random masking therefore risks allocating learning capacity to reconstruct elements that the model already understands well, while providing limited pressure to learn the features that truly distinguish the target domain.

Formally, we consider a pretrained model that has been trained on data drawn from a pretraining distribution X_{PT} . The objective is to adapt this model to a target domain distribution X_T using only unlabeled data, without observing task labels Y . The central challenge is determining how the self-supervised objective used during continued pretraining should be structured so that the model acquires representations that capture the semantic characteristics unique to the target domain. Rather than treating all tokens as equally informative, the key problem is identifying which elements of the input best reflect the difference between the target domain and the original pretraining distribution.

This perspective motivates the study of *difference-aware masking strategies*, where the learning signal during continued pretraining emphasizes the components of the input that most strongly distinguish the target domain. By focusing reconstruction on domain-specific or task-relevant content, the training objective can encourage the model to refine representations that are most critical for downstream reasoning and understanding in that domain.

8.1.2 Reinforcement as Pretraining Objective

The development of LLM has largely relied on scaling along training-time compute (Kaplan et al., 2020; Hoffmann et al., 2022a) and test-time compute (Zhang et al., 2025a). While training-time scaling increases parameters and data through next-token prediction, and test-time scaling (OpenAI, 2024a) enhances reasoning via extended inference, recent works highlight that pretrained models often struggle to realize further gains after post-training (Gandhi et al., 2025a; Wang et al., 2025). The effectiveness of reinforcement learning (RL) on top of supervised fine-tuning (SFT) remains inconclusive: some studies suggest that SFT reduces entropy and hinders RL (Chu et al., 2025; Chen et al., 2025a), while others report improvements from RL applied to heavily SFT’ed models (Liu et al., 2025c). These mixed findings underscore the difficulty of disentangling the relationship between pretraining, SFT, and RL, especially given the lack of transparency around pretraining corpora.

From a broader perspective, this pipeline differs from how learning typically occurs in natural systems. Humans rarely learn by being shown the correct answers for every action. Instead, learning emerges through trial and error: individuals act, observe the consequences, and gradually adjust their behavior. In LLMs, however, this form of learning from experience—what we formalize as RL—appears only at the final stage of development, long after pretraining has shaped the model’s representations. This raises a natural question: if RL can refine reasoning at the end of the pipeline, could introducing experience-based signals earlier help models develop reasoning-friendly representations during pretraining?

Since “learning from experience” is a very broad idea, to keep it within scope, the thesis focuses on a narrower question: *can introducing experience-based learning earlier in the pipeline improve the emergence of reasoning compared to standard pretraining and post-training RL?* To keep this investigation tractable, the initial experiments deliberately focus on controlled settings. We first examine whether replacing standard next-token prediction with a next-token reasoning objective leads to improved reasoning ability during pretraining. If true, we then aim to study whether the gain in early pretraining is sustained after heavy SFT and RLVR. This will consolidate the hypothesis that early experience-based learning unlocks true reasoning potential that is sustained throughout the LLM development cycle.

Base models already exhibit nontrivial reasoning abilities, yet current practice largely postpones the refinement of these skills until alignment stages. Building on recent work in self-reflection and cognitive scaffolding (AI et al., 2025b; Gandhi et al., 2025a), we investigate *reasoning incentive pretraining*—an approach where models are periodically guided to reflect, self-correct, or rank intermediate reasoning steps during pretraining itself. Rather than replacing next-token prediction, this approach interleaves experience-driven signals with standard language modeling to encourage reasoning to emerge as part of representation learning. Dong et al. (2025) takes a step in this direction by reframing next-token prediction as a reasoning task optimized with RL. However, their method required a heavily instruction-tuned model to function under simple rule-based rewards, leaving open the fundamental question of whether reasoning-based pretraining is feasible *earlier* in development, before instruction-following ability has been established. Addressing this gap, our proposed work aims to explore how pretraining and post-training can be unified into a single reasoning-centric paradigm that strengthens intrinsic reasoning skills and yields more generalizable cognitive scaffolds.

Chapter 9

DIFFERENCE-MASKING: Choosing What to Mask in Continued Pretraining

The self-supervised objective of masking-and-predicting has led to promising performance gains on a variety of downstream tasks. However, while most approaches randomly mask tokens, there is strong intuition that deciding *what to mask* can substantially improve learning outcomes. We investigate this in continued pretraining setting in which pretrained models continue to pretrain on domain-specific data before performing some downstream task. We introduce DIFFERENCE-MASKING, a masking strategy that automatically chooses what to mask during continued pretraining by considering what makes a task domain *different* from the pretraining domain. Empirically, we find that DIFFERENCE-MASKING outperforms baselines on continued pretraining settings across four diverse language-only and multimodal video tasks.

9.1 Task Overview

Inspired by the distributional hypothesis in the language domain (Harris, 1954), masking is a self-supervised learning (SSL) objective in which a model attempts to reconstruct hidden portions of data from the surrounding context. Masking has enabled breakthrough performance on tasks from a variety of domains, such as language, vision, and speech (Devlin et al., 2019a; Li et al., 2021; Hsu et al., 2021; Ericsson et al., 2022), motivating interest in researching how masking strategies influence representation learning in SSL. Masked prediction has recently been applied to adapt pretrained models to specific downstream tasks by continuing to pretrain models on in-domain unlabelled data Dery et al. (2023). Masking in this continued pretraining setting been shown to be particularly effective when the target domain differs substantially from the pretraining domain (Gururangan et al., 2020).

While prior work has studied how the *amount masked* influences model learning (He et al., 2022), most masking approaches randomly choose which parts of the data to mask. Although it is understudied in SSL, deciding *what to mask* is a critical component in human education (Pajares & Miller, 1997; Bjork & Linn, 2006). Educators designing “fill-in-the-

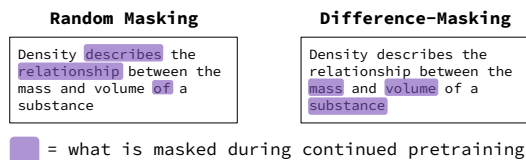


Figure 9.1: DIFFERENCE-MASKING automatically selects *what to mask* based on what makes the task domain *different* from the pretraining domain, enhancing model learning on the end task.

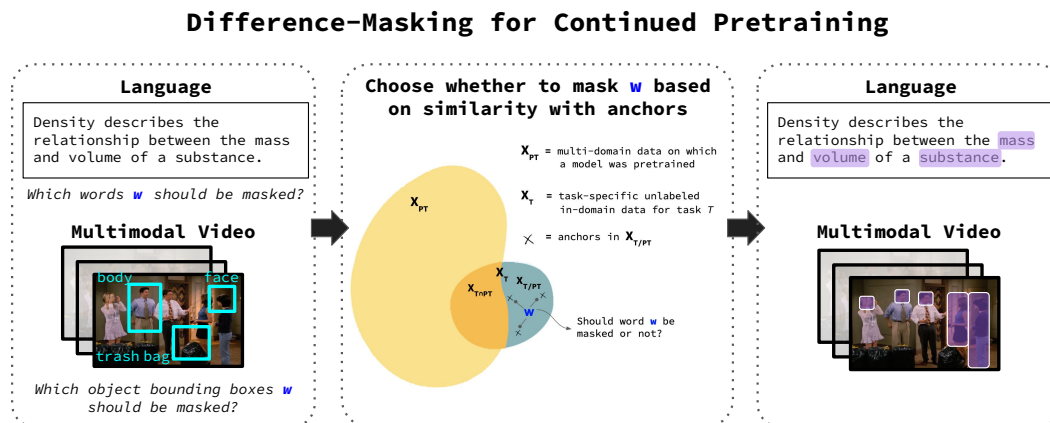


Figure 9.2: DIFFERENCE-MASKING: an approach to choosing what to mask during continued pretraining that prioritizes masking concepts that make the target domain different from the pretraining domain. DIFFERENCE-MASKING does this by first selecting *anchor topics* relating to the downstream task, and then by masking w words or bounding boxes based on their similarity to those anchor topics.

blank” assessments for students must decide what content to mask in order to effectively assess student understanding of a domain (Bae & Lee, 2018). For example, in a real-world “fill-in-the-blank” chemistry test, a teacher might choose to mask domain-specific words (“density”, “silicon”) to assess student learning, instead of masking domain-irrelevant words (“example”, “process”).

In this chapter, we propose DIFFERENCE-MASKING, a novel approach for automatically selecting *what to mask* during continued pretraining. Our strategy first identifies *anchors* that describe what makes a target domain different from the pretraining domain and then determines what to mask during continued pretraining based on similarity to those anchors.

In experiments spanning four diverse language-only and multimodal video datasets (ACL-ARC, ChemProt, TVQA, and Social-IQ), we find that DIFFERENCE-MASKING outperforms strong baselines, supporting our hypothesis that *masking based on what is different* about a task provides strong representation for continued pretraining. We provide intuitions to explain the strong performance of DIFFERENCE-MASKING, along with extensive analyses and ablations to better understand the performance of our method. Our code is publicly available.

9.2 Methodology

This section describes the motivation and implementation of DIFFERENCE-MASKING: our self-supervised method to determine a masking strategy for continued pretraining. The overall process is depicted visually in Figure 9.2.

9.2.1 Problem Setting

We are given a model which has been pretrained on multi-domain data drawn from domain distribution X_{PT} (e.g., a model such as RoBERTa pretrained on a large multi-domain corpus). We are interested in how to adapt this pretrained model to a specific target domain X_T without observing task labels Y .

Continuing pretraining on X_T has emerged as a popular solution approach to this problem (Gururangan et al., 2020; Dery et al., 2023).

9.2.2 Motivation and Notation

If the masking objective is used to train models to learn word representations (Harris, 1954; Devlin et al., 2019a), a natural question emerges: which words is it most important that our models learn to represent? We believe that this question may be important to effectively continue pretraining on specialized domains. We expect that continued pretraining can benefit from a masking strategy that considers what makes a task-domain different.

This leads to the intuition behind DIFFERENCE-MASKING: to train on what makes a target domain different from the pretraining domain. For example, in a corpus about chemistry we would expect that the task of masking and predicting words strongly related to chemistry such as “molecule” will lead to better learning outcomes than words such as “analysis”, which could be related to chemistry in addition to many other domains.

Formally, we term $X_{T \cap PT}$ as the concepts likely to appear in both X_T and X_{PT} (e.g., “analysis”), and we term $X_{T/PT}$ as the concepts that make the domain X_T different from X_{PT} (e.g., “molecule”). With this notation, we can now express our intuition in terms of mutual information with the downstream task Y : we intuit that concepts common in X_T but uncommon in X_{PT} (i.e., in $X_{T/PT}$) share higher mutual information with the task label than concepts found in both domains ($X_{T \cap PT}$) do:

$$I(X_{T/PT}; Y) > I(X_{T \cap PT}; Y) \quad (9.1)$$

The goal of DIFFERENCE-MASKING then is to learn representations during masking that capture the information unique to the domain ($X_{T/PT}$) which is more relevant for the downstream task.

9.2.3 Our Approach: DIFFERENCE-MASKING

To learn masked representations that capture the information unique to the domain ($X_{T/PT}$), our proposed DIFFERENCE-MASKING approach proceeds in two steps:

1. **Finding difference anchors:** We first determine which words are most commonly found in domain X_T and *not* commonly found in general domains X_{PT} . We term these words **difference anchors** that summarize the concepts unique to X_T .

-
2. **Masking based on differences:** Using these difference anchors, we determine the likelihood that each word should be masked based on its similarity to these difference anchors. We sample from this probability distribution to decide what to mask during MLM continued pretraining.

These steps are explained in detail in the following subsections.

9.2.4 Finding Difference Anchors: TF-ICF

Our goal is to determine a set of corpus-level difference anchors that are representative of the differences between the pretraining domain X_{PT} and the task domain X_T . Since our goal is to design a simple yet effective method for finding these differences, we use of a modified version of the widely used TF-IDF scoring function from the field of statistical NLP Jones (1972). TF-IDF determines the ratio of how frequently a word appears in a *document* compared to how frequently the word appears in *other documents in a corpus*. Because we are attempting to find words that make a target *corpus* X_T different from general pretraining *corpora* X_{PT} , the score of a word is highest when it appears frequently in our corpus (X_T) and infrequently in the multi-domain pretraining corpus (X_{PT}). We denote our approach as **TF-ICF** for term-frequency, inverse-*corpus*-frequency, expressed by the following scoring function:

$$\text{TF-ICF}(w_i) = \frac{\text{freq}(w_i, X_T)}{\text{freq}(w_i, X_{PT})} \quad (9.2)$$

To effectively capture word frequencies in the general distribution of the English Language used for pretraining (X_{PT}), we use unigram counts derived from the Google Web Trillion Word Corpus (Brants & Franz, 2006; Norvig, 2009).

We score all words in X_T with this metric and choose the top K as anchors A to represent the domain, where K is a hyperparameter of our method. We analyze the impact of this hyperparameter in Section 9.4.3.

9.2.5 Masking Based on Differences

DIFFERENCE-MASKING then masks words based on similarity to anchors A . Formally, we define similarity between a word w and an anchor word A_k as the cosine similarity of the words' BERT (Devlin et al., 2019a) embeddings.

$$\text{sim}(w, A_k) = \cos(\text{BERT}(w), \text{BERT}(A_k)) \quad (9.3)$$

In order to choose words to mask, we generate probability distribution α over the words in the sentence x to represent the probability that each word should be masked. We determine the weight α_i of each word w_i by calculating its similarity score with the *most similar* anchor word in A (we explore other strategies in our experiments). This value is normalized over the length of the sequence to ensure the probability distribution sums to 1.

$$\alpha(w_i) = \frac{\max_{k \in K} \text{sim}(w_i, A_k)}{\sum_{j=1}^N \max_{k \in K} \text{sim}(w_j, A_k)} \quad (9.4)$$

DIFFERENCE-MASKING then masks terms by sampling from distribution α without replacement, and the model attempts to reconstruct the masked terms from the surrounding context.

Multimodal Implementation of DIFFERENCE-MASKING To apply our method to the visual domain, we draw on work from the vision community in which visual representations are grouped at the object level (Baradel et al., 2018; Sajjadi et al., 2022) and use object labels (e.g. person, car...etc) from a state-of-the-art object detector (Wang et al., 2021; Zhang et al., 2016a) to calculate similarity with the anchor words. A detailed description of our implementation of DIFFERENCE-MASKING in the multimodal setting can be found in Appendix 14.2.

Masking Strategy	Language-Only		Multimodal	
	ACL-ARC	ChemProt	Social-IQ	TVQA
Random Masking (Word)	62.05 _{2.21}	81.90 _{0.51}	-	-
Random Masking (Token)	63.74 _{1.97}	82.82 _{0.23}	69.05 _{0.52}	73.75 _{0.31}
MST (Li et al., 2021)	65.61 _{0.13}	83.17 _{0.17}	68.37 _{0.49}	81.14 _{0.30}
AttnMask (Kakogeorgiou et al., 2022)	66.30 _{1.67}	83.53 _{0.56}	70.18 _{0.71}	81.57 _{0.12}
DGA (Ke et al., 2023)	67.20 _{0.27}	70.67 _{0.30}	-	-
Selective Masking (Gu et al., 2020)	69.06 _{1.80}	82.94 _{0.47}	-	-
EntityBERT (Lin et al., 2021)	71.09 _{0.25}	82.04 _{0.40}	-	-
Salient Span (Cole et al., 2023)	71.94 _{0.58}	82.41 _{0.21}	-	-
DIFFERENCE-MASKING	74.04 _{2.01}	83.94 _{0.39}	71.37 _{0.58}	81.73 _{1.13}

Table 9.1: We find that DIFFERENCE-MASKING outperforms strong baselines in both the language and multimodal experimental settings. We note that our entirely self-supervised method also outperforms Selective Masking, which uses labelled data to inform its masking strategy. Values are average results over five trials, subscripts are standard deviations.

9.3 Experimental Settings

Our experiments evaluate whether DIFFERENCE-MASKING’s masking strategy leads to performance improvements on challenging language-only and multimodal video understanding tasks. We follow the experimental setting from (Gururangan et al., 2020), in which unlabelled data from the downstream task domain is used for continued pretraining before eventually performing downstream task finetuning. This is a popular SSL setting because it represents a computationally-feasible way to test the effectiveness of self-supervised representation learning methods (e.g. without recreating a pretrained model), and it is realistic to modern approaches which rely heavily on pretrained models (Dery et al., 2023).

Experiments are performed to allow each model to learn as long as needed during continued pretraining, only stopping when validation error increases (early-stopping). Each result is averaged across five random seeds. Hyperparameter settings and data preprocessing details can be found in Appendix 14.1.

9.3.1 Datasets

Language-only Datasets As in Gururangan et al. (2020); Dery et al. (2023), we conduct experiments with the **ChemProt** dataset (Kringelum et al., 2016), a relation classification task that uses chemistry documents. ChemProt is a low-resource classification task with a large amount of in-domain unlabeled data, making it a realistic setting in which SSL is helpful in continued pretraining.

We also conduct experiments with the **ACL-ARC** task (Jurgens et al., 2018), a citation intent task based on the ACL Anthology Reference Corpus (Bird et al., 2008) used in continued pretraining experiments in (Gururangan et al., 2020). We use train, validation, and test splits for both datasets from (Dery et al., 2023; Gururangan et al., 2020).

Multimodal Datasets We also experiment on continued pretraining for two challenging multimodal video understanding tasks. **TVQA** (Lei et al., 2018) is a dataset containing 21,792 videos from 6 American television shows and questions and answers related to the videos. Each question is paired with 5 answer choices (one correct answer and 4 incorrect answers), and corresponding video, audio, and subtitles.

Social-IQ (Zadeh et al., 2019) contains 1,250 videos of social situations and questions and answers pertaining to the videos. Each question has corresponding video, audio, and subtitles. We use the train, validation, and test splits from the publicly available datasets.

We use performance metrics consistent with prior work (Gururangan et al., 2020; Dery et al., 2023): F1 score for ACL-ARC and classification accuracy for ChemProt, TVQA, and Social-IQ.

9.3.2 Baseline Methods

Random Masking Most masking approaches choose tokens or words to mask with a uniform random probability (Devlin et al., 2019a; Yang et al., 2019). We consider both the token-level and word-level approaches in our experiments. Formally, the probability α_i that word or token x_i in a sequence of length N will be masked in random-masking is

$$\alpha_i = \frac{1}{N} \tag{9.5}$$

AttnMask (Kakogeorgiou et al., 2022) is a *domain-agnostic* token-based masking approach in which the likelihood of masking a given token is proportional to how attended-to that token is by the [CLS] token, averaged across the different heads of the transformer. Formally, this approach can be seen as defining a function g_{att} which takes in model f_θ , sequence of tokens x , and index i and outputs how attended-to token x_i is.

$$\alpha_i \propto g_{att}(f_\theta, x, i) \tag{9.6}$$

MST (Li et al., 2021) is an approach very similar to AttnMask, except that it masks “non-essential regions”, effectively corresponding to an inverse weighting based on the attention of the model to the token x_i .

$$\alpha_i \propto g_{att}(f_\theta, x, i)^{-1} \tag{9.7}$$

Selective Masking Gu et al. (2020) chooses tokens to mask based on whether adding each token will improve downstream task accuracy as measured by the difference between the downstream task performance when using the full sequence x versus using only the sequence up to and including the token x_i . Notably, this approach *uses downstream task labels to guide the choice of mask* in continued pretraining, whereas DIFFERENCE-MASKING is *self-supervised*.

$$\alpha_i \propto P(y | x) - P(y | x_{[:i]}) \quad (9.8)$$

DGA [Ke et al. \(2023\)](#) is another relevant work that proposes a masking strategy for NLP model adaptation. However, unlike the methods described above, DGA chooses which *attention heads* to mask instead of choosing which *tokens* to mask, assigning importance to attention heads based on the gradient of the loss between the model’s representations of two differently-masked versions of the same input. Additionally, DGA encourages the model to learn integrated representations of the target domain and general knowledge using a contrastive loss.

EntityBERT [Lin et al. \(2021\)](#) masks tokens based on whether they are part of “entities”, as defined by a domain-specific named-entity-recognition (NER) model. The original paper uses the PubMedBERT model, trained originally on the clinical domain. We also implement **Salient Span Masking** ([Guu et al., 2020](#)), which in this case is the same as the EntityBERT approach applied only to mask a single word in the sentence. To apply the approach to the ChemProt and ACL-ARC domains requires NER models effective in those domains. For ChemProt we used the BioBERT model [Lee et al. \(2019\)](#) fine-tuned in NER task with BC5CDR-chemicals [Li et al. \(2016\)](#) and the BC4CHEMD [Krallinger et al. \(2015\)](#) corpus and for ACL-ARC we used the popular SciBERT model ([Beltagy et al., 2019](#)).

9.3.3 Experimental Methodology

Language-only We reproduce the experimental setting from AANG ([Dery et al., 2023](#)), which employs a pretrained 110M RoBERTa_{base} model with two heads: one for continued pretraining and one for the downstream task. Our hyperparameters and other detailed configuration notes are described in Appendix 14.1.

Multimodal We conduct our multimodal experiments using a strong pretrained model: MERLOT-Reserve ([Zellers et al., 2022](#)), a large multimodal transformer pretrained with a contrastive multimodal prediction objective on a dataset of 20 million Youtube videos.

To experiment with masking strategies in the multimodal setting, we continually pretrain a 200M MERLOT-Reserve_{base} model by masking-and-predicting visual patches. We evaluate the learned representation quality by freezing the model and finetuning only the linear classifier layer on the downstream task following ([Wilf et al., 2023](#))’s methodology.

A detailed description of our implementation of DIFFERENCE-MASKING in the multimodal setting can be found in Appendix 14.2, and our hyperparameters can be found in Appendix 14.1.

9.4 Results and Discussion

9.4.1 Comparison with Baseline Approaches

Our experiments compare our proposed DIFFERENCE-MASKING with established baselines including Random Masking (at the word and token level), AttnMask ([Kakogeorgiou et al., 2022](#)), MST ([Li et al., 2021](#)), Selective Masking ([Gu et al., 2020](#)), DGA ([Ke et al., 2023](#)),













ACL-ARC Dataset		ChemProt Dataset	
Frequently Masked Words	ACL Tracks	Frequently Masked Words	Task Labels
 system	Dialogue and Interactive Systems	 activity	activator
 model	Language Grounding	 inhibited	inhibitor
 language	Information Extraction	 inhibitor	agonist
 information	Machine Translation	 cells	substrate
 translation	Machine Learning for NLP	 increased	downregulator
 learning		 human	antagonist

Figure 9.3: The most frequently masked words chosen by the DIFFERENCE-MASKING algorithm across the ChemProt and ACL-ARC tasks. We find that for the ChemProt dataset, the masks we find automatically through unlabelled data partially recover the end task labels.

EntityBERT (Lin et al., 2021), and Salient Span Masking (Cole et al., 2023). The results are summarized in Table 9.1. We find that DIFFERENCE-MASKING shows strong results compared to baselines across language-only and multimodal video understanding tasks.

Notably, our approach demonstrated superior performance on the ACL-ARC dataset with an accuracy of 74.04%, a marked improvement over the random token baseline (63.74%) and a substantial improvement over the best baseline (Salient Span Masking, 71.94%). Our approach also surpassed Selective Masking (69.06%). This is surprising because Selective Masking uses downstream task labels to inform its masking strategy whereas DIFFERENCE-MASKING is self-supervised.

Results on the ChemProt dataset are also encouraging, showing that DIFFERENCE-MASKING achieves an accuracy of 83.94%, marginally better than all the baselines, including Random Masking (82.82%), AttnMask (83.53%), and EntityBERT (82.04%). Similarly to Selective Masking, the EntityBERT and DGA masking strategies were originally tested on much larger datasets, which may suggest a limitation of these methods in the low-resource continued pretraining setting.

DIFFERENCE-MASKING also demonstrates robust performance in multimodal settings. On the Social-IQ dataset, DIFFERENCE-MASKING achieved an accuracy of 71.37%, outperforming the Random Masking (69.05%), AttnMask (70.18%), and MST (68.37%) methods. We were unable to compare our approach with Selective Masking and EntityBERT on these datasets due to the language-only design of their entity taggers. In contrast, our method is not limited to the language domain, and, in fact, performs well in the multimodal setting. And on the TVQA dataset, DIFFERENCE-MASKING achieved an accuracy of 81.73%, outperforming the Random Masking approach substantially (73.75%) and the AttnMask approach marginally (81.57%).

These results highlight the effectiveness and versatility of the DIFFERENCE-MASKING approach across various language and multimodal datasets.

9.4.2 What is masked?

In this section, we investigate what is masked by DIFFERENCE-MASKING and its link to downstream task performance.

On the ACL-ARC task, we find that the most frequently masked words in the ACL-ARC task had an interesting grounding in human intuition. The ACL-ARC task is a citation intent task on a corpus comprising ACL papers. As the subject of ACL papers can vary widely, comprising multiple sub-domains and research fields, we were curious how DIFFERENCE-MASKING’s masking strategy would handle this domain.

We found that the most frequently masked words *closely-aligned with the ACL paper submission tracks* describing the high-level topic categories for papers. For example, some of the most frequently masked words were “learning”, “information”, “translation”, “semantic”, and “lexical”. These words closely correspond to submission tracks “Machine Learning for NLP”, “Information Extraction”, “Machine Translation”, and “Semantics: Lexical”. Since submission tracks for ACL can be seen as a set of topics that span the space of ACL papers, this supports our hypothesis that masked words chosen by DIFFERENCE-MASKING align with what *makes this domain different*.

On the ChemProt task we also found an interesting pattern in what was masked. The objective of the ChemProt task is to determine a type of relation corresponding to a type of biochemical interaction between entities in the text, where labels include words such as “activation”, “inhibitor”, and “antagonist”. Interestingly, we find that some of the words DIFFERENCE-MASKING chooses to mask most often are *the same words as the labels for the downstream task*. This result is also visualized in **Figure 9.3**. Some of the most-masked words by DIFFERENCE-MASKING are “activity”, followed by “inhibited”, “inhibitor”, and “antagonist”. This is a fascinating result because it suggests that *in masking what makes the ChemProt domain unique*, DIFFERENCE-MASKING is determining a self-supervised objective that is *highly similar to the downstream task* without accessing the downstream task labels.

In the multimodal setting we also find an interesting grounding of how DIFFERENCE-MASKING chooses masks in human intuition. Reasoning about social interactions is believed by many psychologists to rely heavily on understanding visual body language cues (De Stefani & De Marco, 2019; Keck et al., 2022). Social-IQ is designed to test these kind of social intelligence capabilities with subtle questions such as “How do the men in the room feel about each other?” and “Do the people in this video feel comfortable about the clown being there?”. In contrast, TVQA tests more general video understanding with question and answer types including those that target visual reasoning about non-human entities and non-visual reasoning from specifically text or audio modalities.

As such, we would expect that our continued pretraining strategy would choose to prioritize masking tokens representing human body language more often in Social-IQ than in TVQA. We found that this was in fact the case. Interestingly, we found that AttnMask baseline also picked up on a similar trend in its attempt to mask based on where attention already focuses, although the trend is much more pronounced in our approach.

The findings in **Table 9.2** demonstrate that DIFFERENCE-MASKING chooses to mask substantially fewer visual tokens corresponding to people than to objects in TVQA, (40%) in comparison to Social-IQ (90%). On the Social-IQ dataset, where the performance differ-

Method	TVQA	Social-IQ
Random	17%	15%
AttnMask	38%	19%
DIFFERENCE-MASKING	40%	90%

Table 9.2: For each method, we analyze what percent of tokens are chosen to be masked from within bounding boxes over people as opposed to objects.

ence is more pronounced over the closest baseline (\uparrow 1.76% over AttnMask), the difference between the proportion of tokens masked from people by these approaches is also most pronounced (90% in DIFFERENCE-MASKING vs 19% in AttnMask).

9.4.3 Sensitivity Analysis

Similarity Function As described in Section 10.2, DIFFERENCE-MASKING determines masking probabilities by comparing the anchor representations to the token representation. Because the token representation is a single vector and the anchors are a group of vectors, similarity can be defined multiple ways. Table 9.1 shows results from the “nearest-neighbor” approach to determining similarity described in Section 9.2.5, motivated by the intuition that a domain can have many *sub-domains* and if a token is close to *any* one of these concepts it should be prioritized for masking. For example, the ACL-ARC corpus has many sub-domains, including the over twenty different submission tracks described in Section 9.4.2. If a paper is about linguistics, it may be important to mask words similar to “language”, whereas if a paper is heavy on ML theory, another anchor might be more appropriate to mask in order to best understand the work.

	ACL-ARC	ChemProt
Centroid	69.02	83.66
Nearest-Neighbor	74.04	83.94

Table 9.3: Ablating DIFFERENCE-MASKING’s anchor-scoring function based on nearest-neighbor and replacing it with one based on similarity with the anchor embeddings’ centroids leads to performance degradation. This provides evidence for our hypothesis that the nearest-neighbor scoring function helps make DIFFERENCE-MASKING robust to anchor selections.

An alternative approach that could be to determine scores by relation to the centroid of the anchor embeddings: in essence, determining whether the token in question is similar to the *anchors on aggregate*. We would expect that this approach would perform similarly to ours on a narrowly-defined dataset such as ChemProt, but substantially differently on a multi-domain dataset such as ACL-ARC. We evaluate this alternative in Table 9.3. We find that the nearest-neighbor strategy does, in fact, outperform the centroid strategy, especially on the ACL-ARC task. This supports our intuition that the nearest-neighbor strategy is particularly helpful when there is a complex or peaky domain.

Number of Anchors In considering the relationship between the anchors and the downstream task, we also investigate how the choice of the number of anchors (K) impacts the downstream performance. We expect that too few anchors will not be expressive enough to determine a strong masking strategy, and too many anchors may begin to overfit to niche concepts that are not representative of the domain. We find that there is indeed a “sweet spot”, and interestingly that it is the same for both datasets: 20. These results are visualized in Figure 9.4.

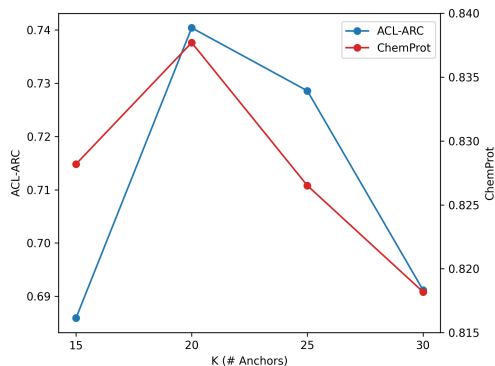


Figure 9.4: Performance on both tasks is best at the hyperparameter $K = 20$ anchors.

9.5 Related Work

Masking relies on the distributional hypothesis, which posits that the meaning of a word can be inferred from its context (Harris, 1954). Masking in NLP has functioned as an effective SSL strategy when training models such as BERT (Devlin et al., 2019a) and XLNet (Yang et al., 2019). Although random masking has been more closely studied in NLP than non-random masking, there are three closely related works to ours from NLP.

EntityBERT Lin et al. (2021) masks tokens based on whether they are part of “entities” recognized by a domain-specific pretrained named-entity-recognizer. Salient Span Masking (SSM) (Guu et al., 2020) is a similar method that uses a named-entity-recognition model to mask out a single entity for the downstream task of open-domain QA. However, these approaches require a domain-specific pretrained entity-tagger, and the masking strategy they determine is the same for any domain to which that tagger is applied. In contrast, DIFFERENCE-MASKING determines what to mask without pretrained entity-taggers, and its masking strategy can change depending on the unlabelled data in the task domain.

Selective Masking Gu et al. (2020) uses data from the downstream task to decide which tokens to mask during continued pretraining by estimating how much each token contributes to improved downstream task performance. It is important to note that Selective Masking uses supervised downstream task labels, whereas DIFFERENCE-MASKING is entirely self-supervised.

Prior work from the vision community has also contributed to an understanding of masking strategies, primarily by using the attention of the model during SSL training to determine what to mask. MST (Li et al., 2021) uses attention maps to determine “non-essential regions” to mask, while AttnMask (Kakogeorgiou et al., 2022) does the opposite by masking the most attended-to regions. Unlike DIFFERENCE-MASKING, these approaches do not take into account domain-specific information when determining their masking strategy. This can be an impediment to performance when the model’s attentions do not already contain information about what is important in a given input sequence.

9.6 Summary

In this paper we introduce DIFFERENCE-MASKING, a method for identifying what makes a target domain unique and using this information to guide a strategy that chooses *what to mask* during SSL continued pretraining. We find that our method outperforms strong baselines across diverse language and multimodal video understanding tasks. We provide a detailed discussion of *what is masked* in DIFFERENCE-MASKING and why our method performs well on various tasks. The cross-task applicability of DIFFERENCE-MASKING supports the effectiveness of our framework for SSL pretraining in language, vision, and other domains.

As described in Section 10.2, DIFFERENCE-MASKING is based on the intuition that it is more beneficial to mask based on what is unique ($X_{T/PT}$) about a downstream task’s domain. However, it is challenging to find what makes a domain unique; therefore, our method is an approximation of $X_{T/PT}$. We believe future work may find it fruitful to investigate additional methods for approximating this, including modifications on the TF-ICF method we proposed. In Section 9.4, we provided intuition, empirical results, and analysis to understand why our method outperformed attention masking baselines by a larger margin on Social-IQ than on TVQA. A broader investigation of why DIFFERENCE-MASKING during pretraining is beneficial by a larger margin to some downstream tasks than to others would be helpful to the community. We believe that self-supervised learning is a promising direction for the machine learning community. This does not discount the salient arguments made about the social and environmental risks of large models (Bender et al., 2021; Strubell et al., 2019). We believe that works such as ours, which study SSL in a resource-constrained context, both increase access to those with limited compute resources and conform to a more environmentally-sustainable way of doing research.

Chapter 10

Reinforcement as Pretraining Objective

The dominant paradigm for training large reasoning models starts with pre-training using next-token prediction loss on vast amounts of data. Reinforcement learning, while powerful in scaling reasoning, is introduced only as the very last phase of post-training, preceded by supervised fine-tuning. While dominant, is this an optimal way of training? In this paper, we present RLP, an information-driven reinforcement pretraining objective, that brings the core spirit of reinforcement learning—exploration—to the last phase of pretraining. The key idea is to treat *chain-of-thought* as an exploratory action, with rewards computed based on the *information gain* it provides for predicting future tokens. This training objective essentially encourages the model to think for itself before predicting what comes next, thus teaching an independent thinking behavior earlier in the pretraining. More concretely, the reward signal measures the increase in log-likelihood of the next token when conditioning on both context and a sampled reasoning chain, compared to conditioning on context alone. This approach yields a verifier-free dense reward signal, allowing for efficient training for the full document stream during pretraining. Specifically, RLP reframes reinforcement learning for reasoning as a pretraining objective on ordinary text, bridging the gap between next-token prediction and the emergence of useful chain-of-thought reasoning.

10.1 Task Overview

Large Language Models (LLMs) pretrained with next-token prediction loss have demonstrated broad utility, but this objective does not explicitly encourage long-range reasoning or integration with world knowledge. Consequently, state-of-the-art models (Guo et al., 2025; Yang et al., 2025a) rely on post-training objectives such as supervised fine-tuning (SFT) and reinforcement learning with human or verified feedback (RLHF, RLAI, RLVR) (Ouyang et al., 2022; Lambert et al., 2024) to induce complex reasoning abilities. In contrast, human comprehension is not a linear token-by-token process, but rather a parallel integration of input with prior knowledge (Baumgaertner et al., 2002; Hagoort et al., 2004; Metzner et al., 2015). Current pretraining lacks such mechanisms, limiting the model’s ability to reason and ground language in world knowledge during learning.

To fill this gap, we propose **Reinforcement Learning Pre-training (RLP)** which treats Chain-of-Thought (CoT) generation as an explicit action taken before predicting each next token. As shown in Fig.10.1, the model first samples an internal thought, then predicts the observed token from the same context augmented with that thought. The training signal is the increase in log-likelihood of the observed token when the thought is present compared to a no-think baseline. This yields a verifier-free and dense reward that assigns position-wise credit wherever thinking improves prediction. Because the signal is defined for ordinary text with teacher forcing, RLP reframes reinforcement learning for reasoning as reinforcement pretraining on the same streams used for maximum likelihood.

Unlike post-training with verifiable rewards, which requires task-specific checkers or curated solutions, RLP is verifier-free: the signal is computed directly from log-evidence under the model and a baseline, allowing uniform application to domain agnostic web-scale text. Compared to reinforcement pretraining via prefix-matching rewards (RPT) (Dong et al., 2025), which uses sparse binary reward and often relies on proxy-model filtering of “easy” tokens, RLP provides a continuous improvement signal at every position and trains on the full documents. This eliminates the need to preselect high-entropy tokens or couple training to a separate small model. Prior RPT demonstrations also depend on distilled checkpoints with strong prior reasoning ability, which clouds whether the method helps base models. RLP is designed to shape thinking in base models by rewarding only those thoughts that measurably help next-token prediction.

This work makes the following key contributions: We introduce **RLP, a verifier-free information-gain objective** that augments next-token prediction by rewarding thoughts in proportion to their predictive utility. We develop a **practical and stable training algorithm** that interleaves reinforcement updates with standard likelihood training via group-relative advantages, a clipped surrogate for thought tokens, and a slowly updated Exponential Moving Average (EMA) baseline. We provide **theoretical guarantees** linking expected reward to reductions in cross-entropy and to a computable lower bound, ensuring both interpretability and tractability. We conduct comprehensive experiments showing that RLP outperforms strong baselines, remains robust after strong post-training, generalizes across diverse corpora, and scales effectively to larger model sizes and hybrid architectures—establishing it as a broadly applicable reinforcement pretraining objective.

10.2 Methodology

We introduce RLP, a pretraining-time procedure that explicitly induces reasoning. As illustrated in Fig. 10.1, RLP inserts a short Chain-of-Thought (CoT) *before* next-token prediction and measures how much that thought improves the model’s log-probability of the observed token relative to a no-think baseline. This improvement, which is a log-likelihood ratio, is a verifier-free, dense reward available at every position in ordinary text corpora. By valuing thoughts in proportion to their predictive benefit, RLP turns reinforcement *pretraining* into learning to think on the same data used for standard next-token training.

Parameterization and roles. We separate the components for clarity:

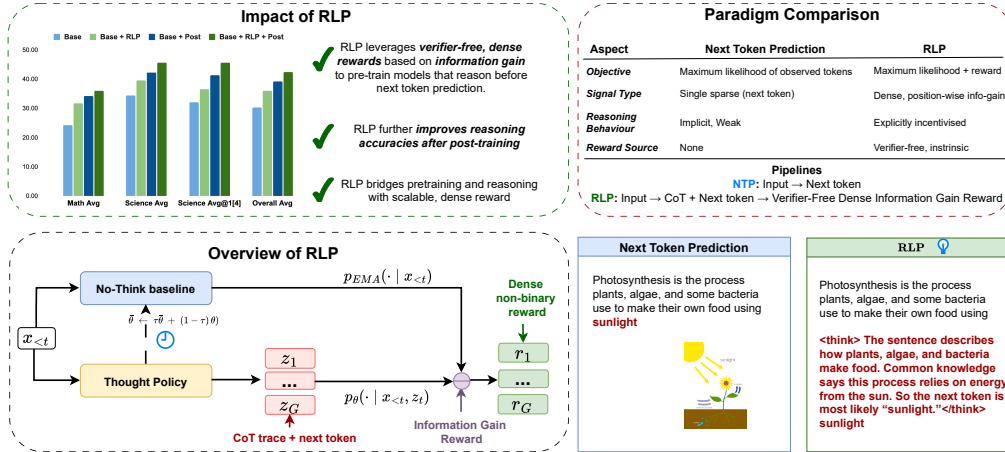


Figure 10.1: **Visualization of the RLP framework.** A chain-of-thought is sampled *before* next-token prediction. Rewards are computed by contrasting the predictor conditioned on the CoT with a *No-think* EMA baseline, yielding a verifier-free, dense signal. We list the advantages of RLP over the traditional pretraining objective (top right) and show the impact after end-to-end training (top left).

- **Thought policy / predictor** $\pi_\theta(c_t | x_{<t})$ and $p_\theta(x_t | x_{<t}, c_t)$ share *exactly the same* network and parameters θ . The network first samples a CoT c_t and then, conditioned on the concatenated prefix $(x_{<t}, c_t)$, scores the next token x_t .
- **No-think baseline** $\bar{p}_\phi(x_t | x_{<t})$ (parameters ϕ) is an EMA teacher of the current network used to score the same token without any CoT channel.

Thus, there is a single model that both *generates* the thought and *predicts* the next token given that thought; the EMA teacher provides the no-think counterfactual.

Classical next-token objective. Given a text sequence $x = (x_0, \dots, x_T)$ and position t , the standard next-token objective for a predictor q_η is

$$\mathcal{L}_{\text{NTP}}(\eta) := \mathbb{E}_{(x_{<t}, x_t) \sim \mathcal{D}} [\log q_\eta(x_t | x_{<t})]. \quad (10.1)$$

For distributions p and q on the next token, we define Cross-entropy (CE) as

$$\text{CE}(p, q) \stackrel{\text{def}}{=} \mathbb{E}_{x \sim p} [-\log q(x)]. \quad (10.2)$$

Using $p^*(\cdot | x_{<t})$ for the data distribution over x_t , maximizing equation 10.1 is equivalent to minimizing $\mathbb{E}_{x_{<t} \sim \mathcal{D}} [\text{CE}(p^*, q_\eta(\cdot | x_{<t}))]$. We include equation 10.1 only for context as our training *does not* include a standard NTP loss term. Instead, RLP optimizes an information-gain objective defined below and updates parameters *only through the tokens of the sampled thoughts*.

10.2.1 Reasoning as an action

RLP augments next-token prediction with a sampled thought. At each position t , the policy draws a latent CoT random variable

$$z_t \sim \pi_\theta(\cdot | x_{<t}),$$

and we write c_t for its realization. The network then predicts x_t with the *reasoned* scorer $p_\theta(\cdot | x_{<t}, c_t)$. As a no-think counterfactual we use $\bar{p}_\phi(\cdot | x_{<t})$, the EMA teacher queried on the same context without providing the CoT.

EMA teacher instantiation and schedule. We instantiate the EMA teacher to match the current model on the *first* batch ($\phi \leftarrow \theta$), and thereafter update it *after* each optimizer step via

$$\phi \leftarrow \tau \phi + (1 - \tau) \theta, \quad \tau = 0.999.$$

This choice makes \bar{p}_ϕ a *moving counterfactual* that is (i) *current* enough to provide informative comparisons and (ii) *intentionally lagged* to mitigate reward hacking. If the baseline were frozen, the counterfactual would drift too far from the evolving model; if it tracked the model without lag, the log-likelihood ratio would collapse toward zero and invite degenerate strategies. The post-update averaging yields a one-step-lagged, smoothed teacher that stabilizes training.

10.2.2 Information-gain reward

With teacher forcing on the next token, define the reasoned and baseline log-evidence

$$S_{\text{pred}}(c_t) := \log p_\theta(x_t | x_{<t}, c_t), \quad (10.3)$$

$$S_{\text{EMA}} := \log \bar{p}_\phi(x_t | x_{<t}). \quad (10.4)$$

The *information-gain* reward is the log-likelihood ratio

$$r(c_t) := S_{\text{pred}}(c_t) - S_{\text{EMA}}, \quad (10.5)$$

which compares the reasoned scorer with a no-think baseline on the observed next token. Rewards are computed under teacher forcing for each t . When updating the policy, we *treat* $r(c_t)$ as a constant with respect to θ (no backpropagation through p_θ or \bar{p}_ϕ); see §10.2.4.

10.2.3 Expected improvement identity

Proposition 1 (CE reduction). *For any fixed $(x_{<t}, c_t)$,*

$$\mathbb{E}_{x_t \sim p^*} [r(c_t)] = \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})) - \text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)).$$

where $p^*(\cdot | x_{<t})$ is the data distribution over x_t . Maximizing the expected reward therefore maximizes the predictive usefulness of the thought for the next token.

Proposition 2 (Lower bound via marginalization over thoughts). *Let $\pi_\theta(z_t | x_{<t})$ be the distribution over CoTs and define the collapsed predictor*

$$\tilde{p}_\theta(x | x_{<t}) = \mathbb{E}_{z_t \sim \pi_\theta(\cdot | x_{<t})} [p_\theta(x | x_{<t}, z_t)].$$

Then for any realized x_t ,

$$\mathbb{E}_{c_t \sim \pi_\theta} [S_{\text{pred}}(c_t)] \leq \log \tilde{p}_\theta(x_t | x_{<t}), \quad \text{and} \quad J(\theta) = \mathbb{E}[r(c_t)] \leq \mathbb{E} \left[\log \frac{\tilde{p}_\theta(x_t | x_{<t})}{\bar{p}_\phi(x_t | x_{<t})} \right].$$

The CoT-conditioned objective is thus a computable lower bound on the improvement one would obtain after marginalizing thoughts. Refer to §14.3.1 of the appendix for the proofs of the propositions.

10.2.4 RLP objective and optimization

RLP optimizes the thought policy to produce thoughts that *increase* predictive evidence. Our training *does not* include the standard next-token loss in equation 10.1. Instead, we optimize only the information-gain objective

$$\max_{\theta} J(\theta) = \mathbb{E}_{x_{<t} \sim \mathcal{D}} \mathbb{E}_{c_t \sim \pi_{\theta}(\cdot | x_{<t})} [r(c_t)], \quad (10.6)$$

or, equivalently, we *minimize* the negative information-gain loss $\mathcal{L}_{\text{IG}}(\theta) = -J(\theta)$. Gradients are applied only to the *thought tokens*; $r(c_t)$ is treated as a constant (no backpropagation through p_{θ} or \bar{p}_{ϕ}).

Group-relative baseline (inclusive mean with correction). To reduce variance, for each context we sample $G \geq 2$ thoughts $\{c_t^{(i)}\}_{i=1}^G$ and use a corrected inclusive mean baseline. Let

$$\bar{r} = \frac{1}{G} \sum_{j=1}^G r(c_t^{(j)}).$$

We define the advantages

$$A^{(i)} := \frac{G}{G-1} \left(r(c_t^{(i)}) - \bar{r} \right), \quad \text{with no gradient propagated through } \bar{r}. \quad (10.7)$$

This multiplicative factor removes the $(1 - \frac{1}{G})$ shrinkage inherent to the inclusive mean, yielding an unbiased estimator with low variance.

Per-token importance ratios and clipped surrogate. We update the log-probability of the *thought* tokens with a clipped surrogate. Let $\ell_u^{(i)}$ be the u -th token in $c_t^{(i)}$ and $\text{prefix}_u^{(i)} = (x_{<t}, \ell_{1:u-1}^{(i)})$. With behavior parameters θ_{old} used to sample the thoughts, define the per-token importance ratio

$$\rho_u^{(i)} = \exp \left(\log \pi_{\theta}(\ell_u^{(i)} | \text{prefix}_u^{(i)}) - \log \pi_{\theta_{\text{old}}}(\ell_u^{(i)} | \text{prefix}_u^{(i)}) \right).$$

We write $\text{clip}(\rho; 1 - \epsilon_{\ell}, 1 + \epsilon_h)$ for elementwise clipping and denote stop-gradient by $\text{sg}(\cdot)$. The surrogate loss is

$$\mathcal{L}_{\text{clip}}(\theta) = -\mathbb{E} \left[\frac{1}{|c_t^{(i)}|} \sum_u \min \left(\rho_u^{(i)} \text{sg}(A^{(i)}), \text{clip}(\rho_u^{(i)}; 1 - \epsilon_{\ell}, 1 + \epsilon_h) \text{sg}(A^{(i)}) \right) \right]. \quad (10.8)$$

10.2.5 Reward properties and guarantees

Does thinking actually help? The reward $r(c_t)$ is positive exactly when the model that used the sampled thought assigns higher probability to the observed next token than the EMA baseline that did not think. In expectation over the data distribution, this equals the reduction in cross-entropy between the reasoned scorer and the no-think baseline (Prop. 1).

Positionwise credit at every step. Since the task is next-token prediction, the reward is computed independently at each position t as

$$r(c_t) = \log p_{\theta}(x_t | x_{<t}, c_t) - \log \bar{p}_{\phi}(x_t | x_{<t}).$$

Credit is attached exactly where the thought changes predictive probability, yielding one scalar per token and removing the need for a learned value function or any external verifier.

Algorithm 1 RLP for next-token prediction with information gain

- 1: **Inputs:** dataset \mathcal{D} , group size $G \geq 2$, clipping $(\epsilon_\ell, \epsilon_h)$, EMA decay $\tau \in (0, 1)$, learning rate η .
 - 2: **Model:** a single network with parameters θ used both as (i) thought policy π_θ and (ii) reasoned predictor p_θ ; EMA baseline \bar{p}_ϕ .
 - 3: **Initialization:** mark ϕ as uninitialized.
 - 4: **while** training **do**
 - 5: Set the behavior snapshot $\theta_{\text{old}} \leftarrow \theta$. ▷ used for the current sampling pass
 - 6: Sample minibatch $\{(x_{<t}^{(b)}, x_t^{(b)})\}_{b=1}^B \sim \mathcal{D}$.
 - 7: For each b , sample G thoughts $c_t^{(b,i)} \sim \pi_{\theta_{\text{old}}}(\cdot | x_{<t}^{(b)})$ with $|c_t^{(b,i)}| \geq 1$.
 - 8: **if** ϕ is uninitialized **then**
 - 9: $\phi \leftarrow \theta$ ▷ lazy init of EMA teacher
 - 10: Compute baseline log-evidence (teacher forcing, no grad) $S_{\text{EMA}}^{(b)}$ as per equation 10.3.
 - 11: Compute reasoned log-evidence $S_{\text{pred}}^{(b,i)}$ and rewards $r^{(b,i)}$ as per equation 10.3 and equation 10.5.
 - 12: Group baseline $\bar{r}^{(b)}$ and $A^{(b,i)}$ (inclusive mean with correction; sg is stop-grad) as per equation 10.7.
 - 13: Per-token importance ratios and clipped surrogate for $\ell_u^{(b,i)}$ with prefix $\text{prefix}_u^{(b,i)}$:
 $\rho_u^{(b,i)} = \exp\left(\log \pi_\theta(\ell_u^{(b,i)} | \text{prefix}_u^{(b,i)}) - \log \pi_{\theta_{\text{old}}}(\ell_u^{(b,i)} | \text{prefix}_u^{(b,i)})\right)$.
 $L_{\text{clip}}^{(b,i)} = -\frac{1}{|c_t^{(b,i)}|} \sum_u \min\left(\rho_u^{(b,i)} \text{sg}(A^{(b,i)}), \text{clip}(\rho_u^{(b,i)}; 1 - \epsilon_\ell, 1 + \epsilon_h) \text{sg}(A^{(b,i)})\right)$.
 - 14: Policy update on thought tokens:
 $\mathcal{L}(\theta) = \frac{1}{BG} \sum_{b=1}^B \sum_{i=1}^G L_{\text{clip}}^{(b,i)}$, $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$.
 - 15: EMA update of baseline: $\phi \leftarrow \tau \phi + (1 - \tau) \theta$.
 - 16: **Output:** trained policy/predictor (shared θ) and EMA baseline ϕ .
-

Putting it all together. Algorithm 1 composes the above pieces into a single training loop. Specifically, multiple thoughts are sampled per position and information-gain rewards are computed against a moving EMA counterfactual. Group-relative advantages are formed and the shared network is updated *only* on the thought tokens via the clipped surrogate in equation 10.8. In this case, the improvements originate from learning to generate thoughts that systematically raise predictive evidence.

10.3 Experimental Setup

We experiment with QWEN3-1.7B-BASE (Yang et al., 2025a) and then scale our experiments to a larger NEMOTRON-NANO-12B-V2 (Nano, 2025) model.¹

RLP. We apply RLP on a diverse set of datasets across two settings: (i) *SFT-style reasoning corpora*, including a math-centric set (OmniMath (Gao et al., 2024a)) and mixed math + general-reasoning sets (OpenThoughts (Guha et al., 2025), NEMOTRON-CROSTHINK (Akter et al., 2025)); and (ii) *general-purpose pretraining corpora*, covering academic papers (ACAD), math textbooks (Math-Text), and open-ended web pages QA pairs from Common Crawl

¹Details about hyper-parameters for each of the below phases and the prompt used for RLP can be found in Appendix 14.5.

Benchmark	$\mathcal{M}_{\text{base}}$	\mathcal{M}_{CPT}	\mathcal{M}	$\mathcal{M}_{\text{base +Post}}$	$\mathcal{M}_{\text{CPT +Post}}$	$\mathcal{M +Post}$
AIME25	2.25	3.96	5.02	5.32	5.89	7.05
MATH500	48.45	57.52	58.48	61.92	62.70	64.30
GSM8K	54.16	72.85	74.48	78.22	78.70	80.50
AMC23	25.94	31.25	31.25	35.00	34.38	36.50
Minerva	15.30	19.03	21.19	25.30	26.10	27.80
MMLU	50.08	41.95	56.14	58.36	59.00	61.50
MMLU@1[4]	44.85	40.00	52.18	56.00	58.53	61.00
MMLU-Pro	28.17	27.81	34.62	37.85	39.92	42.40
MMLU-Pro@1[4]	23.95	24.61	30.80	36.53	38.49	41.30
GPQA	25.25	26.26	28.28	30.93	29.27	33.33
GPQA@1[4]	27.52	24.75	27.02	31.52	30.01	34.97
Math Avg	24.35	30.77	31.74	34.29	34.63	36.03
Science Avg	34.50	32.01	39.68	42.38	42.73	45.74
Science Avg@1[4]	32.11	29.79	36.67	41.35	42.34	45.76
Overall	30.32	30.85	36.03	39.34	39.90	42.51

Table 10.1: Quantitative benchmarks for Qwen3-1.7B-Base, showing the impact of RLP. Shaded columns indicate RLP variants; “Post” indicates SFT + RLVR post-training.

(Web-Crawl)(Nano, 2025). We train with RLP for 1B input tokens using general pretraining corpora (\mathcal{D}_{PT}) to evaluate its effect in an end-to-end LLM pretraining pipeline. We denote this model as \mathcal{M} . Note that theoretically RLP can be applied for every token in a document but in our experiments we randomly select one token per document. Hence, the number of tokens for which the reward is applied is far less than 1B.

Continuous Pretraining. To ensure compute equivalent comparison with \mathcal{M} , we do continuous pretraining on the base model denoted by $\mathcal{M}_{\text{base}}$ with the same tokens used in RLP. We denote this model as \mathcal{M}_{CPT} . This serves as an additional baseline for our experiments.

Post-Training. All models undergo a SFT stage on OpenThoughts data (Guha et al., 2025). To further enhance, we apply Reinforcement Learning with Verifier Rewards (RLVR) using MATH dataset (Hendrycks et al., 2021d). This two-stage post-training pipeline provides an evaluation framework to verify that gains from RLP persist under strong alignment, while also revealing how much additional improvement can be achieved through subsequent post-training. For consistency, all models are trained with identical SFT and RLVR recipes, ensuring that any observed differences in downstream accuracies can be attributed to the pretraining condition ($\mathcal{M}_{\text{base}}$ vs \mathcal{M}_{CPT} vs \mathcal{M}).

10.3.1 Evaluation Metrics

We conduct a thorough benchmark assessment using a series of tasks using NeMo-Skills².

Math Reasoning (MATH AVG). We consider four diverse math benchmarks : GSM8K (Cobbe et al., 2021b), MATH-500 (Hendrycks et al., 2021e), Minerva Math (Lewkowycz et al., 2022b), AMC23. We report Pass@1 average of 8 runs for these.

²<https://github.com/NVIDIA/NeMo-Skills>

Benchmark	$\mathcal{M}_{\text{base}}$	\mathcal{M}	$\mathcal{M}_{\text{base}} + \text{Post}$	$\mathcal{M} + \text{Post}$
MATH500	79.95	78.68	83.47	87.05
GSM8K	72.31	85.98	94.22	94.90
AMC23	70.63	57.19	62.19	75.00
Minerva	22.61	39.48	40.76	42.78
MMLU	54.12	78.76	73.55	78.17
MMLU@1[4]	48.01	79.48	75.23	77.90
MMLU-Pro	24.16	53.13	61.78	67.38
MMLU-Pro@1[4]	27.13	55.76	73.21	66.96
GPQA	25.25	39.90	41.41	48.00
GPQA@1[4]	22.47	48.86	52.15	49.62
Math Avg	61.38	65.33	70.16	74.93
Science Avg	34.51	57.26	58.91	64.52
Science Avg@1[4]	32.54	61.37	66.86	64.83
Overall	42.81	61.32	65.31	68.09

Table 10.2: Quantitative benchmarks for NEMOTRON-NANO-12B-v2, showing the impact of RLP. Shaded columns indicate RLP variants; “Post” indicates SFT + RLVR post-training.

Science Reasoning (SCIENCE AVG). For conceptual science and specialized knowledge, we evaluate on MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024c), and the graduate-level STEM benchmark GPQA-Diamond (Rein et al., 2024). For science benchmarks, we report the average greedy and Pass@1 scores from 4 runs (SCIENCE AVG@1[4]).

10.4 Results

Table 10.1 reports the performance of QWEN3-1.7B-BASE under different pretraining and post-training objectives. First, RLP consistently outperforms both the $\mathcal{M}_{\text{base}}$ and \mathcal{M}_{CPT} across nearly all benchmarks, with especially strong gains on reasoning-heavy tasks such as AIME25 and MMLU-Pro. We see that \mathcal{M} is relatively on average 19% and 17% better than $\mathcal{M}_{\text{base}}$ and \mathcal{M}_{CPT} respectively. This highlights the effectiveness of dense, verifier-free reinforcement signals for instilling reasoning capabilities during pretraining. Second, the benefits of RLP persist even after strong post-training (SFT + RLVR). While all models improve after post-training, \mathcal{M} achieves the highest scores with the overall average substantially higher than both $\mathcal{M}_{\text{base}}$ by 8% and \mathcal{M}_{CPT} by 7% relatively. This indicates that RLP establishes robust reasoning foundations that are not washed out by downstream alignment but instead compound with post-training. We observe particularly large gains in science domains, with $\mathcal{M} + \text{Post}$ achieving +3 points over $\mathcal{M}_{\text{CPT}} + \text{Post}$. This trend suggests that RLP is not limited to mathematical reasoning but also generalizes effectively to other domains. The ability to strengthen performance in science benchmarks highlights that RLP fosters a broader class of multi-step explanation-driven reasoning skills, moving beyond domain-specific improvements and pointing toward a more versatile foundation for reasoning in LLMs. Overall, the results demonstrate that RLP not only induces reasoning ability during pretraining but also synergizes with post-training, leading to models with stronger and more durable reasoning abilities than those trained with next-token prediction or continuous pretraining.

Scaling Model Size and Architecture We further scale RLP to NEMOTRON-NANO-12B-V2 (Nano, 2025) ($\mathcal{M}_{\text{base}}$), a hybrid Mamba-Transformer language model of 12B parameter size. In this comparison we take an intermediate checkpoint of NEMOTRON-NANO-12B-V2 trained till 19.8 trillion tokens and apply RLP for 250 million tokens only. $\mathcal{M}_{\text{base}}$ on the other hand is trained for 20 trillion tokens. In addition, we employ an identical post-training pipeline (SFT \rightarrow RLVR), mirroring the setup used for QWEN3-1.7B-BASE in Table 10.1. The results as shown in Table 10.2 confirms that, regardless of model size and families, RLP not only yields a very large improvement at the base stage (Overall 42.81% to 61.32%, a 43% relative gain) but that these gains persist and continue to compound after strong post training. After SFT + RLVR, the RLP trained model improves from 61.32% to 68.09%, maintaining a clear margin over the compute matched baseline (65.31%). The largest relative gains are in scientific reasoning as the Science Avg rises from 34.51% to 57.26% at the base stage and further to 64.52% after post training, compared to 58.91% for the continuously pretrained baseline. This pattern closely mirrors our findings on Qwen3, and demonstrates that RLP scales effectively both to larger parameter counts and to a different architecture family, while remaining compatible with strong downstream alignment. Furthermore, we validate that RLP scales effectively to larger backbones by applying it to QWEN3-14B-BASE, where it improves the overall average from 60.66% to 65.00% after training on 1B tokens, with particularly strong gains in scientific reasoning. Full results for this scaling experiment are provided in the Appendix 14.7.

RPT Comparison Following the experimental setup in RPT (Dong et al., 2025), we trained $\mathcal{M}_{\text{base}}$ on both RPT and RLP methods for one epoch under tokens and flop matched compute budgets before evaluating on our benchmark suite. In the token matched setting, we trained both models on Omni-MATH (Gao et al., 2024a) using the same number of input tokens. As we apply RLP to a single token per document while RPT is applied on multiple tokens per document, the number of target tokens for which reward is applied is substantially larger for RPT.

Conversely, for the flop-matched training, both models are trained on Nemotron-CrossThink (as detailed in Appendix 14.7) for one epoch on the same data, and we ensure that the number of target tokens for which reward is applied is same in both training runs. As summarized in Table Table 10.3, under *Token-Matched* setting, RLP achieves uniformly higher aggregates: *Math Avg* improves by an absolute +2.12% (+4.5% relative), *Science*

Avg by +1.19%(+3.3% relative), and *Overall Avg* boosts by +1.66% (+4.0% relative). In addition, under the flop matched setting, the improvement is even more prominent. RLP achieves a 20.12% relative improvement on average over RPT. Methodologically, RPT applies reinforcement only to tokens pre-selected by an auxiliary assistant via entropy filtering and optimizes a sparse, binary next-token correctness signal that ignores the CoT content, limiting where the signal can be applied. In contrast, RLP evaluates each sampled CoT by

Model	Math Avg	Science Avg	Avg
<i>Token-Matched</i>			
\mathcal{M}_{RPT}	47.50	35.88	41.69
\mathcal{M}	49.62	37.07	43.35
<i>Flop-Matched</i>			
\mathcal{M}_{RPT}	36.66	34.38	35.68
\mathcal{M}	45.95	38.76	42.86

Table 10.3: Token- and flop-matched comparisons of RLP and RPT using a QWEN3-1.7B-BASE model.

Model	Dataset	Type	Math Avg	Science Avg	Science Avg@1[4]	Avg
$\mathcal{M}_{\text{base}}$	-	-	35.96	34.50	32.11	34.19
\mathcal{M}_{CPT}	NEMOTRON-CROSTHINK [170M]	Equal Input Token	37.11	35.76	32.15	35.01
	NEMOTRON-CROSTHINK [6B]	Equal FLOPs	43.90	37.74	32.47	38.04
	$\mathcal{D}_{\text{PT}}[1\text{B}]$	PT Data Mix	45.34	32.14	29.33	35.60
\mathcal{M}	OmniMath [170M]	SFT	46.48	40.27	37.54	41.43
	OpenThoughts [170M]		47.64	40.84	35.88	41.45
	NEMOTRON-CROSTHINK [170M]		49.76	42.54	37.78	43.36
	ACAD [170M]	General	47.68	40.59	36.87	41.71
	Math-Text [170M]		48.07	40.46	36.32	41.62
	Web-Crawl [170M]		48.87	40.75	36.77	42.13
	$\mathcal{D}_{\text{PT}}[1\text{B}]$	PT Data Mix	46.35	39.68	36.67	40.90

Table 10.4: **RLP across diverse corpora.** RLP trained on six SFT-style and general-purpose datasets yields consistent gains, indicating transferable reasoning from mixed/open-ended data.

the information gain it provides for the observed next token and updates at all positions without an auxiliary filter which yields consistently better averages under the matched setting above. Crucially, this dense, per-token information-gain reward supplies richer credit assignment than RPT’s sparse binary signal and, in our matched experiments, empirically yields better performance.

10.5 Ablations

Does RLP provide generalizable improvements across diverse corpora? A key advantage of RLP is its scalability to large, diverse corpora, unlike RLVR, which relies on small, curated reasoning datasets and raises concerns about generalizability. Prior work (Chen et al., 2025c; Setlur et al., 2025) highlights the need for complex reasoning corpora to sustain RL improvements, but such datasets are costly to curate and impractical at pretraining scale. For these ablations, we apply RLP to QWEN3-1.7B-BASE for 200 steps—utilizing 170M input tokens—holding the rest of the setup fixed.

As illustrated in Table 10.4, RLP delivers consistent gains across all corpus families, eliminating concerns that RL based pretraining only benefits curated reasoning data. Relative to $\mathcal{M}_{\text{base}}$ average improves by 7-9% with strongest gains on NEMOTRON-CROSTHINK (SFT-style) and Web-Crawl (general-purpose corpora). Unlike prior work (Akter et al., 2025), where RL gains were limited to math and weakened under mixed data, RLP achieves simultaneous improvements across all benchmarks, demonstrating genuine cross-domain transfer. Even on purely non-reasoning general corpora such as web-crawl, RLP extracts a reasoning signal that scales with data diversity (Appendix 14.7). Table 10.4 illustrates that unlike prior work (Liu et al., 2025b; Zhou et al., 2025b), RLP can be applied to any data format like academic papers, textbooks, web-crawl as well as SFT style data. Overall, RLP is scalable, domain-agnostic pre-training augmentation that enhances both reasoning and accuracy.

Does the improvement sustain under compute equivalent baselines? A critical question is whether RLP’s gains stem from its unique RL-based pretraining or simply higher compute. Standard next-token pretraining quantifies compute by input tokens, but RLP adds rollout

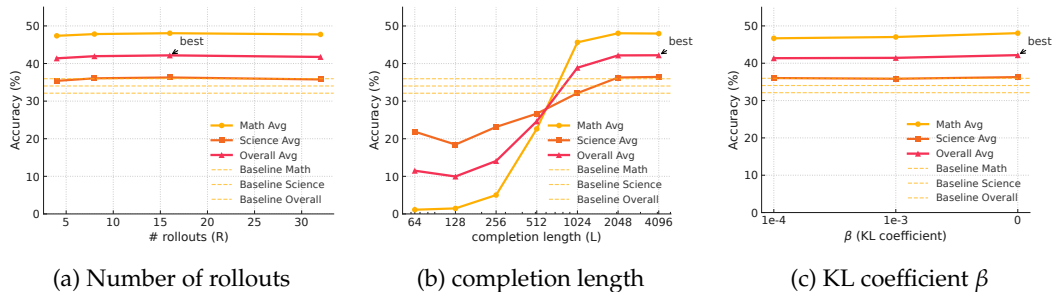


Figure 10.2: **Ablations on Qwen3-1.7B.** Curves report Math/Science/Overall averages. Dashed lines mark the base model.

costs not captured by this metric. For fair comparison, we evaluate against \mathcal{M}_{CPT} baselines under: (a) equal Input Tokens Seen and (b) equal total Compute FLOPs. RLP is fixed to $T_{\text{inp}} = 170\text{M}$ tokens; the token-matched \mathcal{M}_{CPT} [170M] continues pretraining on 170M tokens (Input Token), while the FLOP-matched budget corresponds to 6B tokens for CPT (\mathcal{M}_{CPT} [6B])(see Appendix 14.6).

In Table 10.4, \mathcal{M} outperforms \mathcal{M}_{CPT} trained on the same 170M tokens and maintains a clear advantage even against a compute-matched \mathcal{M}_{CPT} exposed to 6B tokens (35 \times more data). Despite this disparity, RLP achieves a 5.3% gain on average (compare \mathcal{M}_{CPT} NEMOTRON-CROSTHINK [6B] vs \mathcal{M} NEMOTRON-CROSTHINK [170M]), with consistent improvements across math and science benchmarks. These results show that RLP’s gains stem not from more efficient use of compute, not larger budgets, validating the effectiveness of our approach.

Is comparable to CPT with high-quality reasoning data? High-quality reasoning corpora have shown to substantially boost base model reasoning ability when used in continuous pretraining (CPT) or mid-training (Wang et al., 2025; Gandhi et al., 2025a). This raises the important question of whether CPT can match or even surpass under such favorable conditions. To investigate this, we conduct CPT on both reasoning-centric, NEMOTRON-CROSTHINK and general pretraining (\mathcal{D}_{PT}) datasets, each using 170M tokens. Our results in Table 10.4 show that even with high quality reasoning data, consistently outperforms CPT by a significant margin. Specifically, \mathcal{M} outperforms \mathcal{M}_{CPT} , showing an average gain of 8% on NEMOTRON-CROSTHINK and 5% on pre-training data mix (\mathcal{D}_{PT}) on 1B tokens. These results highlight two key insights. First, while CPT benefits from reasoning-dense corpora, it remains sensitive to domain skew—evident in the weak science accuracy on \mathcal{D}_{PT} —whereas generalizes more evenly across disciplines. Second, the consistent margin by which outperforms CPT, even in the presence of high quality reasoning data, underscores that the gains of are not merely due to data quality but stem from the algorithmic design itself. This reinforces the conclusion that provides a generalizable mechanism for leveraging reasoning data during pretraining, complementing rather than being overshadowed by high-quality corpus selection.

Ablations on rollout count, completion length, and KL weight. Fig. 10.2 visualizes the trends across three settings: (a) rollouts, (b) completion length, and (c) KL. Please look into §14.6 for more detailed numbers and per-task breakdowns. More rollouts help up to

$G = 16$ (Overall 42.17%); $G = 4$ and 8 already reach 41.38% and 41.95%, while $G = 32$ decreases slightly to 41.75% (Fig. 10.2a). Increasing completion length gives the largest gains. Specifically, *overall* rises from 11.50% at 64 to 42.17% at 2048, with *Math/Science* moving from 1.12%/21.88% to 48.06%/36.29% (Fig. 10.2b). Extending to 4096 yields 42.21% at roughly twice the thought budget, so we default to 2048. Furthermore, a KL anchor does not help. Specifically, $\beta = 10^{-4}$ and 10^{-3} give 41.35% and 41.44%, compared to 42.17% at $\beta = 0$, and it also increases memory and step time (Fig. 10.2c). We therefore use $G = 16$, completion length 2048, and $\beta = 0$ in later experiments.

10.5.1 How RLP Impacts the Base Model Representation Across LLM Families

Beyond downstream task gains, we ask *where* in the network RLP alters the model. We compare a base pretrained model and its RLP checkpoint on the same reasoning-oriented prompts and measure how much their internal representations differ at each layer. We run the same set of short reasoning prompts (e.g., “Solve step by step...”) through both models with `output_hidden_states=True`, and compare their hidden states layer by layer.

For each layer, we compute: (1) **Centered Kernel Alignment (CKA)** between the base and RLP hidden states over all prompt positions—CKA measures representation similarity and equals 1 when representations are identical up to an orthogonal transformation; (2) **mean L2 difference** of hidden vectors (base vs. RLP) across positions and samples—this quantifies how far activations have moved after RLP. The figures additionally report complementary probes (attention entropy, overlap, and span; hidden-state variance ratios; per-head CKA summaries; token-level entropy and logit similarity by position), all computed on the same prompts.

We perform this comparison across three model families that differ in scale, routing structure, and sequence-mixing mechanism, allowing us to ask whether RLP leaves the same mechanistic signature in all settings or whether its effect depends on the underlying architecture. **Nemotron-Nano-V2** is a **12B-parameter hybrid dense** model. **Nemotron-Nano-V3** is a **30B mixture-of-experts** model with **3.5B active parameters**. **Nemotron-Super** is a **120B total / 12B active** hybrid **Mamba-attention MoE** model with **multi-token prediction (MTP)** layers.

Attention is frozen; learning is routed elsewhere. In all families, attention entropy, span, overlap, mean $|\text{RLP} - \text{Base}|$, cosine similarity of attention maps, and top- k key overlap are *bit-identical* between Base and RLP (difference zero where applicable, similarity and overlap 1.0). That is a strong structural fact: RLP does not retune *where* tokens attend. Interpretively, the RLP signal is tied to conditional next-token prediction under internal thoughts and an EMA baseline—it does not need to rewrite attention patterns to reuse the same contextual readout the base model already learned. On hybrid stacks (Mamba-attention, MoE), this further suggests that capacity for absorbing RLP updates sits in *non-attention* submodules—state-space blocks, routed experts, layer norms, and output/MTP heads—while the attention operator continues to implement the same discrete routing over keys. The distinction matters for scaling: attention is a shared, highly constrained map over positions; changing it would alter inductive biases globally, whereas RLP appears to specialize the *post-attention* computation and readout.

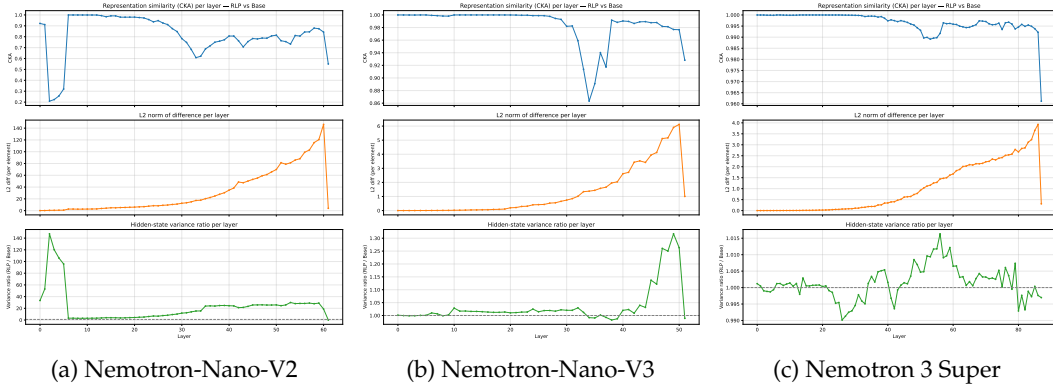


Figure 10.3: **Representation change after RLP, by layer.** Each column is one model family: CKA between base and RLP hidden states (top), mean per-element L2 $\|\mathbf{h}_{\text{RLP}} - \mathbf{h}_{\text{base}}\|_2$ (middle), and hidden-state variance ratio $\text{Var}(\mathbf{h}_{\text{RLP}})/\text{Var}(\mathbf{h}_{\text{base}})$ (bottom). Same prompts and pooling over positions for all panels.

Nemotron-Nano-V2 (12B dense): early shock, preserved middle trunk, explosive deep rescaling. As shown in Figure 10.3a, layer-wise CKA (RLP vs. Base) shows a sharp early “V”: similarity collapses to ~ 0.2 around layer 2, then recovers to ~ 1.0 for layers 6–25, then drifts down through the last third of the network with a terminal CKA ~ 0.55 . The hidden-state variance ratio mirrors the early dip (spike $\sim 145\times$ at layer 2), stays modestly above 1 in the stable band, then plateaus around $20\text{--}30\times$ in deep layers before collapsing at the final layer. The per-element L2 difference stays negligible until \sim layer 26, then grows to a large peak ($\sim 140+$) just before the last layer, where it snaps back toward zero alongside CKA’s final drop. Raw hidden statistics reinforce this: RLP’s mean L2 norm and variance track the base in shallow layers but diverge dramatically in depth (e.g., $\sim 4\times$ larger norm and order-of-magnitude larger variance at the deepest pre-head layer).

What this means architecturally. In a *dense* 12B model, every token traverses the same full stack, so any representational change induced by RLP must be absorbed by the shared trunk rather than isolated within a sparse subnetwork. The result is a highly structured three-phase pattern. First, RLP introduces a strong early perturbation immediately after the input interface, suggesting that reasoning-oriented pretraining changes how the model initially organizes the input signal. Second, the long middle band remains almost perfectly aligned with the base model, indicating that much of the core linguistic processing pipeline is preserved. Third, the deep layers show strong rescaling and growing representational separation, implying that RLP primarily alters the way higher-level features are amplified and recombined near the pre-logit stack. The sharp drop in L2 distance and variance ratio at the final layer, despite a simultaneous drop in CKA, suggests that the final hidden states are brought back to a comparable numerical scale while remaining geometrically distinct. This is most consistent with a change in representational basis or readout alignment at the output boundary, rather than a simple continuation of the deep-layer magnitude growth. More broadly, NanoV2 suggests that in smaller dense models, RLP improves reasoning by injecting a strong early representational perturbation and then exploiting the upper layers to build much larger and more diverse internal features.

Nemotron-Nano-V3 (30B MoE, 3.5B active): a mid-late bottleneck, then expressive finals. The CKA traces in [Figure 10.3b](#) tell a different story from a smaller, dense stack. CKA is ~ 1.0 through roughly layer 30, then falls in a pronounced V-shaped band (minimum ~ 0.86 near layer 34), recovers to ~ 0.98 – 0.99 for layers 38–50, and ends with a modest final dip (~ 0.93). The L2 difference stays near zero until mid depth, accelerates through the last third, peaks (~ 6) at the penultimate layer, then drops at the last layer. The variance ratio hugs 1 for most of the network, then **rises** in the deepest layers (peak ~ 1.32), meaning RLP *increases* activation spread just before the head—matching the per-layer plots where RLP exceeds the base in both mean L2 norm and variance in the final $\sim 15\%$ of depth (e.g., higher peaks at layer 50).

What this means architecturally. MoE training already encourages **specialized subnetworks**; only 3.5B of 30B parameters are active per token. RLP’s effect clustering in a **narrow depth band** (layers 31–37) plus the **final stack** suggests that routing, expert norms, or residual statistics shift where abstract features are composed, while early representation extraction stays shared across experts and matches the base. The late **variance inflation** is not “regularization” in the sense of uniformly shrinking activations—it is **increased channel diversity** in the layers that feed the head, plausibly because different experts or gates amplify distinct directions once the thought policy is enabled. Per-head CKA shows near-zero spread early, elevated spread exactly in the CKA-dip region and a **sharp final-layer spike**—the output-adjacent change is concentrated in a few heads, consistent with sparse, head-specific readout rather than a layer-wide rotation.

Nemotron 3 Super (120B / 12B active, Mamba-attention MoE, MTP): trunk statistics preserved, geometry and readout shift. At the largest scale, CKA remains $\gtrsim 0.999$ through ~ 40 layers, then drifts down gradually into the 0.990 – 0.995 range before a **sharp terminal drop** (CKA ~ 0.96 at the last layer) (Shown in [Figure 10.3c](#)). L2 difference accumulates from mid depth, peaks (~ 4) near the penultimate layer, then collapses (~ 0.3) at the final layer—a decoupling between **subspace similarity** (CKA still “high” in absolute terms) and **coordinate-wise distance** (large mid-network, small at the last layer). The variance ratio departs only mildly from 1 (brief dip ~ 0.99 , peak ~ 1.016 in a mid-late band). Crucially, **mean L2 norm and variance of hidden states are visually identical** for RLP and Base at every layer: RLP does not globally rescale activation energy the way it does in NanoV2.

What this means architecturally. A wide MoE with Mamba and MTP has many degrees of freedom to implement a new conditional predictor *without* changing first- or second-moment statistics of hidden states—e.g., by **rebalancing experts, shifting state-space evolutions, or retuning MTP/auxiliary heads** while layer norms hold scale fixed. The progressive CKA erosion with depth is then best read as a **cumulative subspace drift**: small per-layer rotations compound, even when norms are matched. The **final-layer CKA cliff** plus exploding per-head CKA spread aligns with MTP and the primary LM head living at the boundary of the stack: RLP likely reallocates **which heads** implement which n-gram shortcuts or logits, even when intermediate magnitudes look unchanged. In other words, Super exhibits **surgical** representation change at scale—geometry and specialized units move, not raw activation size. The pattern is consistent with RLP modifying task-relevant and reasoning-related computation. That CKA drops and mean diff increases most in the second half of the network suggests that RLP refines the way the model represents reasoning and problem-solving in those layers, rather than changing low-level feature extraction. The

pronounced change at the last layer may reflect a shift in how the model maps its final representation to next-token predictions (e.g., toward more “reasoning-heavy” solutions favored by the reward). Together with improved downstream reasoning performance, this analysis supports the view that RLP improves reasoning by *reorganizing high-layer representations* rather than by altering early layers.

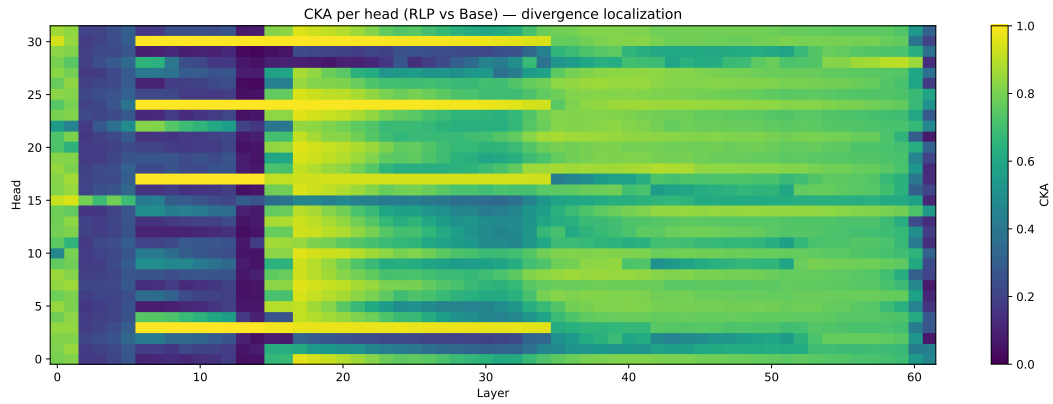
Per-Head CKA Reveals Where Divergence Is Localized. Layer-averaged curves can mask *intra-layer* structure: a modest mean CKA might reflect all heads moving a little, or a few heads moving a lot while the rest stay fixed. [Figure 10.4](#) resolves this by plotting CKA (base vs. RLP) for each attention head and layer jointly.

Across families, the two-dimensional pattern orders cleanly by scale and routing. **NanoV2** shows the richest spatial texture: vertical bands where many heads shift together alternate with horizontal “rails” of heads that stay near $\text{CKA} \approx 1$ for long depth spans, so divergence is neither layer-uniform nor head-uniform but *structured*—consistent with aggressive repurposing of specific channels in a dense width where those edits remain visible before they compound upstream. **NanoV3** is qualitatively different: the map is nearly uniform yellow outside a *thin* mid-depth stripe and the output-adjacent column, so the MoE can confine representational change to a small depth interval and to whichever heads lie on the activated expert pathways for these prompts. **Super** is almost saturated yellow wall-to-wall, with only faint mottling mid-stack and modest cooling on a handful of heads near the top—so the same training signal leaves little *resolved* head-level footprint in CKA space even though layer-mean CKA still drifts; many small coordinated adjustments across width and modules substitute for the sharp stripes seen in the smaller model.

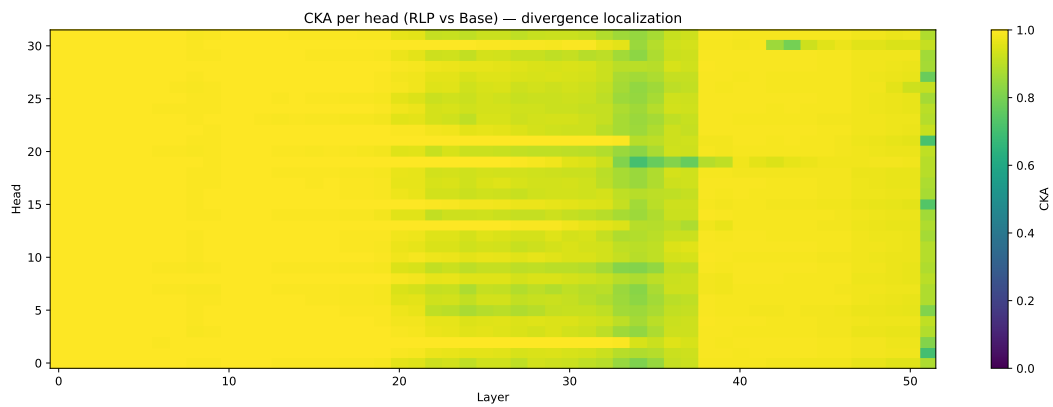
Read alongside the layer-wise plots above, the heatmaps support a single scale-dependent picture: visible head-level rewiring is most pronounced when width is fully updated on every forward pass (dense 12B), becomes *spatially delimited* when sparsity allows updating only some routes (MoE), and largely *blends into the background* at the largest hybrid scale, where reasoning gains can ride on diffuse geometric drift and readout retuning without a dramatic layer \times head mosaic.

Cross-family interpretation. Taken together, the corrected figures suggest that the effect of RLP is strongly dependent on both scale and architecture. In the smaller dense NanoV2 model, RLP induces large-amplitude changes, first at the input interface and later in the deep trunk, indicating that reasoning requires broad internal rescaling when capacity is limited, and computation is fully shared. In NanoV3, which introduces sparse expert routing, the effect becomes more localized: most of the network remains aligned with the base, but a narrow mid-layer band and the final stack are selectively modified. In Super, where capacity, modularity, and auxiliary prediction structure are all much richer, RLP leaves global hidden-state moments almost untouched and instead manifests primarily as gradual subspace drift plus output-boundary specialization.

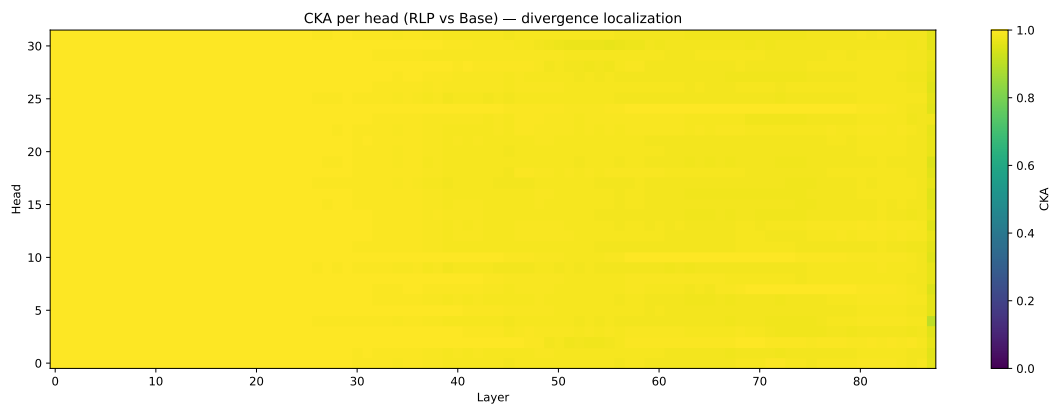
The broader implication is that reasoning-oriented pretraining does not imprint a single universal signature. Smaller dense models appear to require large representational deformation to internalize reasoning behavior, whereas larger and more modular architectures can absorb the same training signal through increasingly targeted and geometrically subtle changes. This supports the view that RLP is not simply making activations larger or more



(a) Nemotron-Nano-V2



(b) Nemotron-Nano-V3



(c) Nemotron 3 Super

Figure 10.4: **CKA per head (base vs. RLP)**. Each panel is a layer \times head heatmap (yellow \approx identical subspaces, purple/blue \approx strong divergence). Same prompts and pooling as Fig. 10.3.

variable. Rather, it is altering where and how the model encodes reasoning, with the form of that alteration determined by the architecture’s available degrees of freedom.

10.6 Related Work

Next-Token Prediction. Next-token prediction is the standard pretraining objective for LLMs: predict the next word from prior context (Shannon, 1951; Bengio et al., 2003). Scaling it with Transformers (Vaswani et al., 2017b) enabled landmark and state-of-the-art systems (Radford et al., 2018; Brown et al., 2020b; Smith et al., 2022; Bi et al., 2024; Nano, 2025; Yang et al., 2025a). Anticipating tokens across corpora induces syntactic, semantic, and pragmatic structure that transfers broadly. Alternatives include masked language modeling (Devlin et al., 2019b) and span corruption (Raffel et al., 2020b), but next-token prediction remains dominant for its alignment with left-to-right generation and strong downstream accuracy across tasks. In this work, we add a verifier-free dense reward during pretraining that leverages reasoning before prediction.

Verifier-Free Rewards in Post-Training. Recent work explores verifier-free rewards. Yuan et al. (2024) uses iterative DPO where, after SFT, the model judges its own candidates to create preference pairs. Liu et al. (2025b) trains with incentive RL on SFT corpora. Zhao et al. (2025) proposes RL from an internal feedback while using the model’s confidence as reward. RLP, in contrast, is a GRPO-style pretraining objective. It operates on any text data including web-crawl, academic papers and SFT datasets and optimizes continuation quality beyond next-token prediction. Because these methods target post-training policies, direct comparisons are not well-posed.

10.7 Summary

We introduce RLP, a reinforcement pretraining objective that rewards chain-of-thought by its information gain for next-token prediction. Unlike traditional approaches that defer RL to post-training, RLP instills reasoning during pretraining, yielding gains that persist and compound after alignment. Experiments across datasets, domains, and architectures show that RLP consistently outperforms compute-matched baselines and scales efficiently to large hybrid models, establishing reinforcement pretraining as a principled and general alternative to likelihood-only training.

Part IV

Discussion and Future Directions

Chapter 11

Conclusion

11.1 Summary of Contributions

This thesis investigates how to build reasoning-centric learning systems by studying the problem from three connected perspectives: the quality of the data used to teach reasoning, the stage of training at which such data is most effective, and the learning objectives that can make reasoning emerge earlier and more robustly in the training pipeline. Across these directions, the central claim of this thesis is that strong downstream reasoning does not arise from scale alone. It depends on carefully designed supervision, principled placement of that supervision across training phases, and training objectives that reward models for acquiring representations and behaviors aligned with reasoning from the beginning.

The first major contribution is a systematic study of what makes synthetic data useful for reasoning in generative and multimodal systems (Part I). Rather than treating all synthetic data as equally beneficial, the following chapters show that structure, grounding, and informational content strongly influence what the model learns; cross-domain high-quality data, in addition, paves the way for gains that transfer across multiple benchmarks. In chapter 3, we proposed the VisReas dataset and a model that executes step-wise pseudocode with verification before answering, so that VQA models must verify question-image consistency at each reasoning step; the impact is a first step toward reliable VLMS in real-world settings where user instructions may be imperfect, and existing VQA models are shown to be less robust on unanswerable and multi-hop reasoning under this formulation. In chapter 4, we proposed using a single VLM to generate visual representations of text queries (via HTML rendering) and then to solve the problem from both text and the self-generated image, improving math and general-purpose reasoning when the generated image structures information relevant to the solution and reducing reliance on external image-generation APIs. In chapter 5, we proposed converting raw web text into math-informed synthetic *dialogues* (e.g., two professors, teacher-student, debate) with knowledge gaps between participants; pretraining on MIND-OWM (Nemotron-MIND) significantly improves mathematical and logical reasoning over raw or rephrased text, and structured conversations prove more effective than surface rephrasing for multi-hop reasoning. In chapter 6, we proposed a framework to incorporate multi-domain corpora into training via curated sources, MCQ/open templates to control answer space, verifiability filtering, and data-blending strategies, enabling scaling self-learning beyond math so that models trained

with NEMOTRON-CROSSTHINK (Nemotron-CrossThink) generalize better across reasoning tasks and exhibit task-appropriate verbosity, with verifiable rewards in non-deterministic domains. Together, these chapters establish that synthetic data is most valuable when it reflects the latent structure of reasoning rather than merely imitating surface-form answers.

The second major contribution is an investigation of how reasoning data should be distributed across the stages of large language model development. A recurring assumption in recent work is that reasoning is primarily a post-training phenomenon, induced through supervised fine-tuning or reinforcement learning after broad pretraining has already taken place. In contrast, this dissertation studies whether reasoning-oriented supervision can produce more lasting benefits when introduced earlier. The [chapter 7](#) examines how the quality, complexity, and redundancy of reasoning data affect transfer when injected at different phases of the pipeline. This analysis helps answer a practical but underexplored question: whether the same reasoning data should be repeated across phases, diversified across phases, or concentrated in only one of them. The resulting findings offer a more principled account of when reasoning data is most useful and how early exposure can strengthen downstream adaptation.

The third major contribution is a redesign of the learning pipeline itself so that reasoning is not treated only as an outcome of late-stage alignment. In the final part of the thesis in [Part III](#), I explore how pretraining objectives can be modified to encourage reasoning-oriented representations and behaviors before conventional post-training begins. One direction ([chapter 9](#)) studies continued pretraining through the lens of what the model should pay attention to, asking how masking strategies can emphasize the information that most distinguishes a target domain from the model’s original pretraining experience. This outperforms random masking on language and multimodal video tasks (ACL-ARC, ChemProt, TVQA, Social-IQ), showing that masking domain-distinctive content improves adaptation. A second direction ([chapter 10](#)) investigates whether reasoning incentives can be embedded directly into pretraining, allowing models to benefit from intermediate thought-like processes rather than relying exclusively on next-token prediction. This adaptation provides consistent gains over next-token baselines and shows robustness to post-training and scalability across model sizes and corpora. Together, these chapters argue that reasoning emergence can be shaped not only by better data, but also by better objectives that make learning itself more aligned with structured inference.

Empirical gains alone, however, leave open a critical question: *why* does reasoning-oriented pretraining work, and *where* in the network does it act? Without mechanistic grounding, it is impossible to know whether the gains are fragile artifacts of a particular data mix, whether they would survive at larger scale, or how future training recipes should be designed. To answer this, both directions are accompanied by analyses that open the black box of internal network computation using Centered Kernel Alignment (CKA), hidden-state L2 distance, variance ratios, per-head CKA heatmaps, logit divergence, and token-level prediction entropy. For front-loaded reasoning data ([chapter 7](#)), these probes show that the gains are not superficial: the strongest representational divergence concentrates in the early-to-middle layers where the model organizes token sequences into structured intermediate states, specific attention heads are selectively reconfigured for long-range dependency tracking and sub-problem decomposition while general linguistic machinery stays intact, and output-space logit divergence grows progressively with generation length—

confirming that the internal difference is behaviorally consequential precisely where it matters most, deep inside a multi-step reasoning trace. The mechanistic upshot is that post-training is refining a backbone that already encodes latent structure for decomposition, state tracking, and multi-step inference, rather than constructing that structure from scratch—explaining why the gains transfer durably across benchmarks and resist being overwritten by subsequent fine-tuning.

For the RLP objective ([chapter 10](#)), the same probes applied across three architecturally distinct model families—reveal that attention maps are bit-identical between Base and RLP checkpoints in every family, establishing that RLP does not rewrite where tokens attend but instead specializes post-attention computation and readout. The locus and amplitude of change shift systematically with scale: the dense 12B model absorbs RLP through large early perturbations and deep-layer rescaling; the sparse MoE confines change to a narrow mid-layer band via expert routing; and the largest hybrid model achieves the same reasoning gains through gradual subspace drift and output-boundary specialization, with global activation magnitudes nearly untouched. This scale-dependent pattern has a direct implication for practice: reasoning-oriented pretraining does not demand a single fixed recipe, but rather a training signal that different architectures can absorb through their own available degrees of freedom, making RLP a robust and architecture-agnostic foundation for reasoning emergence.

Taken as a whole, the thesis contributes an end-to-end view of reasoning-centric model development. It explains how to construct better reasoning data, when to introduce it for maximum effect, and how to redesign pretraining so that reasoning abilities are supported from the earliest stages of learning. Beyond the individual methods and empirical findings in each chapter, the broader contribution is a unified perspective: reasoning should be treated as a property that is cultivated throughout the learning pipeline, from data design to objective design, rather than as a skill added only after pretraining is complete.

11.2 Future Directions

11.2.1 Data Anatomy in Model Development

Understanding the Structural Anatomy of Reasoning Traces. While the studies in this part of the thesis highlight the importance of structured supervision for reasoning, several open questions remain regarding the internal structure of high-quality reasoning data. Observations from publicly released chain-of-thought corpora suggest that effective reasoning trajectories often contain recurring stages such as hypothesis formation, intermediate solution attempts, verification of partial results, and iterative refinement. These stages resemble how humans approach complex problems, where reasoning unfolds through exploration, validation, and correction. Future work will aim to systematically study these structural patterns and determine how each reasoning component contributes to model learning and generalization. Understanding the anatomy of reasoning traces may provide principled guidelines for constructing training corpora that explicitly teach models how to reason rather than simply predicting answers.

Guided Multi-Step Synthetic Reasoning Generation. A promising direction is the development of guided multi-step frameworks for generating synthetic reasoning data. Instead

of producing a single chain-of-thought explanation, future systems may construct reasoning trajectories through multiple stages, where models first generate hypotheses, attempt solutions based on those hypotheses, evaluate intermediate steps, and refine incorrect reasoning. Such a process would expose models to richer reasoning dynamics, including how alternative strategies are explored and compared. By controlling the structure and sequencing of these reasoning stages, researchers can study how the depth and organization of reasoning traces influence the reasoning capabilities of trained models. This direction may also enable the construction of synthetic reasoning corpora that contain structured exploration and self-correction, properties that are rarely present in conventional question–answer datasets.

Teacher Models for General-Purpose Reasoning Data Generation. Most current approaches for training reasoning models focus heavily on mathematical and coding domains, largely because correctness in these domains can be verified deterministically. However, many real-world reasoning tasks extend beyond such structured settings and involve contextual interpretation, domain knowledge, or heuristic decision making. The Nemotron-CrossThink framework suggests a promising direction by incorporating multi-domain corpora and diverse question formats into reinforcement learning pipelines. By exposing models to reasoning tasks spanning science, humanities, and commonsense reasoning, multi-domain supervision may encourage models to develop adaptable reasoning strategies that generalize across different problem types. Models trained with multi-domain reasoning supervision may also become powerful teacher models for generating synthetic reasoning data. Once a model acquires reasoning abilities across varied domains, it can produce structured reasoning traces that capture different cognitive strategies used to approach complex problems. These teacher models could therefore bootstrap large-scale reasoning datasets for domains where annotated reasoning data is scarce. For example, a reasoning-capable model could generate explanations, intermediate reasoning steps, and verification signals for tasks involving policy analysis, historical reasoning, or scientific hypothesis evaluation. This capability may significantly expand the scope of reasoning datasets beyond the domains where automatic verification is straightforward.

Together, these directions suggest that the next phase of reasoning-centric model development will depend less on increasing token counts and more on designing training data that explicitly encodes the structure of reasoning. Understanding how to construct, scale, and integrate such reasoning data remains a central challenge and a promising avenue for advancing general-purpose reasoning in large language models.

Recognizing Reasoning via Intrinsic-Like Metrics. A natural follow-up question is whether one can recognize that a model is able to do reasoning via some *intrinsic* metric rather than relying solely on downstream task accuracy. AI et al. (2025b) introduce a framework to measure *reflection* (a form of metacognition where the model examines and corrects prior reasoning) during pre-training. They define **explicit reflection** as the production of tokens that explicitly acknowledge and address mistakes in adversarial chains-of-thought. Using a prompt-based classifier to detect such phrases, they report **Explicit Reflection Rate**—the fraction of task instances where the model output exhibits explicit reflection—and show that it correlates strongly with $\log(\text{pre-training compute})$ (e.g., Pearson correlation $\approx 0.76\text{--}0.86$ across tasks). Accuracy on adversarial tasks (implicit or explicit reflection) also increases with pre-training. Thus, the rate of explicit reflection in model generations,

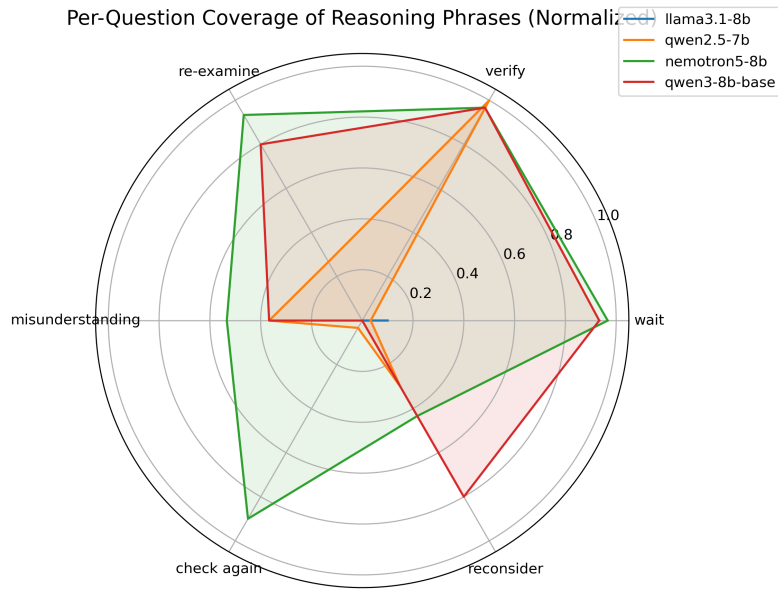


Figure 11.1: **Per-question coverage of reasoning phrases (normalized)** in AIME-25 solutions from four LLMs. Each axis is one phrase (verify, wait, reconsider, check again, misunderstanding, re-examine). Larger area indicates broader and higher use of these phrases. The ranking by coverage aligns with AIME-25 accuracy: Nemotron 5-8B-base > Qwen 3-8B-base > Qwen 2.5-7B-base > LLaMA 3.1-8B-base.

elicited under controlled prompts (e.g., with a “Wait,” trigger), acts as an *intrinsic-style* signal that does not require full task-accuracy evaluation and correlates with both compute and downstream recovery from errors. Table 5 in that paper gives examples of reflection phrases (e.g., “Wait, 42.6 is not a whole number. Let’s solve this step by step...”) that the classifier detects.

We conducted an ablation to test whether **per-question coverage of reasoning phrases** in model-generated solutions correlates with downstream reasoning performance. We collected solutions from four LLMs (LLaMA 3.1-8B, Qwen 2.5-7B, Qwen 3-8B-base, Nemotron 5-8B) on AIME-25 and measured the normalized presence of six reasoning-related phrases in each solution: *verify*, *wait*, *reconsider*, *check again*, *misunderstanding*, and *re-examine*. **Figure 11.1** shows the per-question coverage (normalized) for each model across these six axes as a radar chart. LLaMA 3.1-8B exhibits the smallest footprint—minimal coverage on all phrases—while Nemotron 5-8B shows the largest and broadest coverage, with high presence on *verify*, *wait*, *reconsider*, *check again*, *misunderstanding*, and *re-examine*. Qwen 3-8B-base and Qwen 2.5-7B lie in between, with Qwen 3 showing stronger coverage than Qwen 2.5. This ordering **aligns with downstream AIME-25 performance**: Nemotron has the highest AIME-25 score, followed by Qwen 3, then Qwen 2.5, then LLaMA. Thus, reasoning phrase coverage in model outputs—measurable without running full task accuracy—serves as an intrinsic-like proxy that correlates with competition-level math performance.

Beyond reflection phrases, other work proposes metrics that do not reduce to task accuracy. **PHi** (Prediction of Hidden states) measures how well a model can predict its *own* future hidden states during generation; high unpredictability is taken to indicate novel compu-

tation and reasoning rather than trivial or memorized processing (Zhang et al., 2025b). PHi correlates with task complexity and with the correctness of self-generated reasoning chains. **Geometric and topological** analyses of reasoning traces (e.g., progress and stability of the trajectory in representation space) have been used to distinguish coherent reasoning from hallucination or low-quality CoT. These are more expensive to compute than phrase coverage but offer a representation-level view. Surveys on “beyond accuracy” evaluation of reasoning behavior (Mondorf & Plank, 2024) emphasize that reasoning can be assessed via generation traces, explicit reflection rate, and consistency of strategy, in addition to final-answer accuracy.

There is no single, universally accepted intrinsic metric for reasoning analogous to perplexity for language modeling. However, **explicit reflection rate** (via phrase detection on model outputs under reflection-eliciting prompts) and **reasoning phrase coverage** both correlate with downstream reasoning performance and can be computed without full task-accuracy evaluation. Our AIME-25 ablation shows that models with higher coverage of phrases like “verify,” “wait,” and “re-examine” tend to perform better on AIME-25; LLaMA, with the lowest coverage, has the lowest score. These metrics therefore offer a practical way to recognize and compare reasoning capability across models using only their generated text.

11.2.2 Effective Integration of High Quality Data

Scaling Front-Loading to Frontier Models and Architectures. While our current experiments in chapter 7 demonstrate significant gains on 8B-parameter models, a fundamental question remains regarding whether reasoning-rich pretraining scales linearly with model capacity or eventually reaches a point of saturation. Future work should extend the FLR framework to much larger parameter counts and longer token horizons, exceeding the 1-trillion-token mark. It is critical to test whether the asymmetric principle—which prioritizes reasoning diversity in early phases and high-quality refinement in later stages—holds across diverse architectural paradigms such as Mixture-of-Experts (MoE) or hybrid State Space Models (SSM). Investigating these scaling laws will determine if larger models require a higher proportional density of reasoning tokens to unlock their latent cognitive capabilities, or if their increased capacity allows them to generalize from more sparse logical signals.

Temporal Optimization and the "Critical Period" for Reasoning. Our current work suggests that introducing reasoning data early in the pipeline is beneficial, yet the exact "chronology of learning" remains to be fully mapped. We propose exploring the temporal dynamics of pretraining to identify the optimal injection point for durable reasoning gains using progressive or curriculum-based schedules. This research would involve analyzing "forgetting curves" for reasoning primitives to see if front-loaded logic is "washed out" during the final stages of massive web-scale pretraining. Finding this balance is essential for ensuring that models retain deep structural reasoning without compromising their generalizability or linguistic fluency. Mapping this learning trajectory could lead to more compute-efficient training recipes where logic is solidified before the model undergoes broad stylistic alignment.

Multimodal Front-Loading and Perception-Grounded Intelligence. The principles of front-loading reasoning have primarily been validated within the text domain, but human

reasoning is deeply rooted in physical and sensory grounding. A significant future direction involves investigating whether early multimodal pretraining—integrating vision, audio, and symbolic data—yields the same persistent benefits as text-based reasoning data. We hypothesize that early grounding in visual or symbolic logic, such as geometry, physics simulations, or the "clique-based" visual reasoning discussed in Chapter 2, may create a cognitive scaffold that improves abstract logical deduction in later text-only stages. Examining this cross-modal generalization will be a key step toward developing perception-grounded intelligence, where a model's ability to reason about the world is informed by a fundamental understanding of spatial and relational constraints learned early in its development.

Together, these directions suggest that the next phase of reasoning-centric model development will depend on a sophisticated understanding of when and how to introduce structured logic into the learning pipeline. By optimizing the timing, scale, and modality of reasoning data, we can move closer to creating models that possess intrinsic, durable cognitive capabilities applicable across the full spectrum of human knowledge.

11.2.3 Rethinking Training Pipeline

The methodologies introduced in this part of the thesis represent a fundamental shift from passive data ingestion to active, objective-driven learning. By prioritizing domain-specific uniqueness through Difference-Masking and incentivizing internal deliberation via Reinforcement Learning Pretraining (RLP), we have laid the groundwork for reasoning-centric development. However, several frontier challenges remain in optimizing these algorithms for global-scale training and diverse architectural paradigms.

Refined Approximations of Domain Uniqueness. The current implementation of Difference-Masking rests on the intuition that masking tokens unique to a downstream task ($X_{T/PT}$) provides a superior representation for continued pretraining. While our current approximation using TF-ICF has yielded strong empirical results, determining the precise "essence" of a domain remains an open challenge. Future research should investigate more sophisticated methods for approximating domain uniqueness, such as using semantic density maps or gradient-based importance sampling. Furthermore, a broader investigation into why this strategy yields significantly larger margins of improvement in specific multimodal contexts, such as social intelligence, versus structured video QA, would provide the community with a more nuanced understanding of how specialized "vocabularies of thought" are formed during the adaptation phase.

Scaling Reinforcement Pretraining to Trillion-Token Horizons. While RLP has demonstrated the efficacy of treating "thinking" as an explicit action, our current experimental setup has been limited to a 6B-token scale. A critical next step is to determine whether the verifier-free information-gain objective remains stable when scaled to trillions of tokens. Scaling RLP involves addressing the computational overhead of maintaining an Exponential Moving Average (EMA) baseline and managing the variance of dense rewards across massive, heterogeneous web crawls. Future work will focus on optimizing the parallelization of thought-augmented training to ensure that the "reasoning incentive" can be applied at the same scale as standard maximum likelihood estimation without prohibitive compute costs.

Multi-Token Thought and Prediction Cycles. The current RLP architecture is designed around a single-token prediction cycle, where the model samples a thought to predict the immediate next token. However, human reasoning often involves long-range planning where a single "insight" governs a sequence of subsequent actions or words. Extending RLP to support multi-token prediction—where an internal thought-trace is rewarded based on its ability to improve the log-likelihood of an entire subsequent phrase or block of code—could significantly enhance the model’s coherence in complex tasks. This shift from point-wise prediction to sequence-level integration would allow the model to develop more "macro" reasoning capabilities, reducing repetitive or unstructured elements in the chain-of-thought.

Hybrid Objectives and Training from Scratch. Currently, RLP is primarily utilized toward the end of the pretraining phase to refine existing representations. A transformative direction for the learning pipeline is the investigation of Hybrid Loss Objectives that interleave next-token prediction (NTP) and reward-based signals from the very first step of training. We aim to explore whether RLP can be applied "from scratch," allowing reasoning to emerge as a primary capability rather than a secondary refinement. By balancing the standard cross-entropy loss with a reasoning-incentive reward throughout the entire lifecycle of the model, we may be able to bridge the gap between pretraining and post-training, creating base models that are inherently "thinkers" before they ever encounter an instruction-tuning dataset.

Extending Reasoning Objectives to Sparse Architectures. Finally, as the field moves toward more compute-efficient architectures, it is essential to adapt these redesigned learning pipelines to Mixture-of-Experts (MoEs). Since MoEs activate different sub-networks based on the input context, integrating a thought-augmented objective like RLP presents a unique opportunity: can the model learn to route specific "reasoning experts" more effectively when a task requires deep deliberation? Future work will examine how dense reward signals can be used to regularize expert routing, ensuring that the model’s internal "thought" process is not only accurate but also architecturally efficient.

Extending RLP as a Generalized RL Objective during Post-Training. In [chapter 6](#), we extended RLVR to non-math domains by curating data and applying task-specific templates—MCQ and open-ended with constrained answer length—to *limit answer-space variability* and enable rule-based reward signals. That approach addresses why broader domains are harder than math: in general-purpose reasoning, correctness is often non-deterministic and answer spaces are diverse, so designing a verifiable reward (e.g., exact match or format check) is difficult without constraining the task. Converting or templating open-ended questions into MCQ or short-answer format is one way to make RLVR applicable, but it does not scale to tasks that are inherently open-ended, such as summarization or creative writing, where there is no single "correct" answer to verify. RLP, in contrast, uses a *verifier-free, dense* reward defined at every token: the information-gain reward measures how much the sampled chain-of-thought improves the model’s probability of the observed next token relative to a no-think baseline ([chapter 10](#)). This reward does not require a ground-truth answer or any external verifier; it is derived entirely from the data distribution and the model’s own predictions. As a result, RLP can in principle be applied to *any* text corpus, including domains where RLVR is inapplicable—summarization, long-form generation, creative writing, or open-ended dialogue—simply by rewarding thoughts

that improve next-token prediction on the observed continuation. A promising direction for future work is to extend this objective beyond pretraining: deploying an RLP-style, information-gain-based reward as a post-training RL objective would provide a more *general* approach than templating for verifiability, enabling reinforcement learning on broad domains without reducing every task to MCQ or short-answer format. Whether such a post-training formulation preserves stability and alignment, and how it interacts with existing preference-based objectives, remains to be studied.

Together, these directions aim to unify the training process into a single, reasoning-centric paradigm. By refining what we mask, scaling how we think, and optimizing where we route, we can ensure that the next generation of LLMs possesses the emergent cognitive scaffolds necessary to tackle the world’s most complex challenges.

11.2.4 Reasoning Bases for Long-Horizon and Agentic Tasks

Long-horizon and agentic tasks require more than fluent next-token prediction. They demand persistent goal tracking, decomposition of a problem into subgoals, evaluation of intermediate states, and correction when a partial path fails. The central implication of this dissertation is that these abilities are more likely to emerge when reasoning is cultivated in the base model itself, rather than added only during post-training. Across the thesis, structured reasoning data repeatedly exposes the model to stepwise inference, intermediate validation, conversational decomposition, and refinement. This matters because long-horizon tasks are precisely the settings where each early step changes the quality of later steps. A model that has only learned to imitate short final answers may appear competent on local reasoning benchmarks, yet fail when success depends on keeping a coherent plan over many turns or actions. By contrast, a reasoning-enriched base is trained on data that makes the path to the answer visible: multi-turn dialogues in MIND, multi-domain structured reasoning in Nemotron-CrossThink, and reward-based pretraining in RLP all encourage the model to represent intermediate reasoning states rather than only output endpoints. The thesis explicitly argues that front-loading reasoning data into pretraining creates durable foundations that post-training alone cannot recover, which is exactly the kind of durable internal structure needed for long-horizon and agentic behavior.

Pretraining on Agentic Trajectory Data: From Single-Turn to Multi-Turn Reasoning. This thesis focuses on single-turn reasoning traces—chain-of-thought sequences that move from a problem statement to a solution. Agentic tasks, however, unfold as multi-turn trajectories in which the model alternates between reasoning, acting (invoking tools), observing results, and re-reasoning. The structure of an agentic trajectory is fundamentally different from a chain-of-thought: it includes external observations, partial states, and recovery from unexpected outcomes. We can develop pretraining corpora that include agentic trajectory data—sequences of (thought, action, observation) triples drawn from real or simulated agent interactions—and study whether the FLR asymmetric principle generalizes: does front-loading diverse agentic trajectories into pretraining create similarly durable benefits for multi-turn, tool-augmented reasoning? The Behavior Priming approach (Jin et al., 2026), which endows models with reasoning-centric behaviors via SFT, suggests that process-level supervision during training is more beneficial than outcome-level optimization alone. Applying this principle at the pretraining scale—before any instruction tuning—may yield

a reasoning base that is natively oriented toward agentic behavior rather than treating tool invocation as an add-on. Concretely, this could involve mining open-source agent interaction logs (from platforms like OpenHands¹ or SWE-agent scaffolds), constructing synthetic agentic trajectories using models like ReTool², and studying how the diversity and quality of these trajectories interact with pretraining scale in the same way [chapter 7](#) studies single-turn reasoning data.

Temporal and Critical-Period Analysis for Agentic Capability Development. We have shown the importance of studying the chronology of learning in pretraining—identifying critical periods during which reasoning data yields maximal durable benefit and mapping forgetting curves for reasoning primitives. This direction acquires additional urgency in the agentic setting, where different capabilities (planning, tool use, error recovery, multi-agent communication) may have different critical periods. To determine this, we can train model checkpoints with progressively more agentic trajectory data introduced at different points in the pretraining curriculum; evaluate each on a task suite probing distinct sub-capabilities; identify whether these sub-capabilities have separable critical periods; and determine whether agentic trajectory data introduced early is ‘washed out’ by subsequent web-scale pretraining at a different rate than mathematical reasoning data. This connects to the broader agentic reasoning survey (Wei et al., 2026) finding that self-improving agents generating new training states from inference-time behavior may be the key to continuing capability improvements when training data plateaus.

Multi-Token Thought and Hierarchical Planning for Agentic Decision-Making. As proposed in the previous section, RLP can be extended to multi-token prediction cycles—where an internal thought-trace is rewarded for improving the log-likelihood of an entire subsequent phrase rather than just the immediate next token. This has direct relevance to long-horizon agentic tasks, where human planning is characterized by macro decisions (high-level strategy) that govern many subsequent micro-decisions (individual actions within that strategy). The more interesting future works can be implementing a hierarchical RLP objective that rewards thoughts at two timescales: a fast reward measuring improvement on the next-token prediction (as in the current implementation) and a slow reward measuring improvement on a longer sub-sequence—perhaps the next tool call and its immediate aftermath. This mirrors hierarchical RL frameworks. Recent work on subgoal-driven agent frameworks (Wang et al., 2026a) and explicit lookahead-based planning like FLARE (Wang et al., 2026b) suggests that bridging this timescale gap—evaluating candidate actions by simulating the trajectories they induce—is a key missing ingredient in current reasoning model architectures. The hierarchical RLP objective would embed this multi-timescale structure into pretraining itself.

11.3 Broader Impact

Recent advances in large language models have been driven largely by increases in model scale and training data. While this scaling paradigm has produced impressive gains in language understanding and generation, it has also revealed fundamental limitations in

¹<https://openhands.dev/>

²<https://retool.com>

reasoning, compositional generalization, and robustness across domains. This thesis investigates an alternative perspective: that reasoning capability can be shaped not only by scale, but by the structure of training data and the design of learning objectives. By examining how reasoning supervision interacts with different stages of the model development pipeline—from dataset construction to pretraining and reinforcement learning—this work contributes toward a deeper understanding of how reasoning behaviors can emerge systematically during training.

A central implication of this research is the importance of data design in shaping model capabilities. The studies presented in this thesis suggest that structured reasoning traces, multi-step supervision, and carefully designed training objectives can substantially influence the development of reasoning skills. These findings indicate that progress in reasoning may depend less on simply increasing the size of models and datasets, and more on designing training corpora that explicitly encode problem-solving strategies. The datasets introduced in this thesis, including MIND-OWM and NEMOTRON-CROSTHINK, have already been incorporated into the development of recent reasoning-oriented models such as (NVIDIA, 2025a) Nemotron-H, a hybrid 8B parameter LLM; (NVIDIA, 2025b) Nemotron-Nano-V2, a hybrid 12B parameter LLM; and (NVIDIA et al., 2025) Nemotron-3-Nano-30B-A3B, a 30-billion-parameter mixture-of-experts model with 3.5B active parameters. The consistent improvements observed when training these models with the proposed datasets demonstrate that reasoning-focused data can enhance model capability across a range of architectures and scales, from compact hybrid models to larger MoE systems.

The emergence of stronger reasoning capabilities has important implications for the reliability and transparency of AI systems. Models trained to produce structured reasoning traces are more likely to expose intermediate steps in their decision process, which can help users inspect and verify model outputs. In domains such as education, scientific research, and decision support systems, the ability to trace how a model arrived at a particular conclusion can increase trust and facilitate human oversight. Structured reasoning supervision may therefore contribute to the development of AI systems whose outputs are not only accurate but also interpretable and easier to evaluate. Approaches such as reasoning-focused pretraining, reinforcement learning during pretraining, and structured synthetic data generation demonstrate that reasoning behaviors can be encouraged earlier in the training process. This perspective may influence how future language models are trained, shifting emphasis from purely scaling compute toward understanding the qualitative structure of training data.

Improvements in reasoning ability also broaden the range of tasks that language models can support. Many real-world problems involve multi-step inference, integration of knowledge across domains, or iterative refinement of hypotheses. Training approaches that emphasize reasoning diversity and multi-domain reasoning exposure may enable models to generalize more effectively across problem types. This capability could support applications in areas such as scientific discovery, policy analysis, and complex problem solving, where systems must reason through multiple possibilities before arriving at a conclusion.

The training strategies explored in this thesis also carry implications for the efficiency of AI development. Large-scale models require significant computational resources, and continued scaling raises concerns about environmental cost and accessibility. By demonstrating that reasoning improvements can arise from changes in data structure and learning ob-

jectives, this work suggests alternative pathways for advancing model capability without relying exclusively on increased compute. Methods that leverage structured supervision or reinforcement learning signals during pretraining may help improve reasoning ability within existing computational budgets.

At the same time, the growing reasoning ability of language models introduces new societal challenges. Systems capable of generating detailed reasoning chains and persuasive explanations could potentially be used to produce misleading arguments or highly convincing misinformation. Because reasoning traces can create an impression of logical validity, incorrect or manipulated reasoning may be difficult for non-expert users to detect. Addressing these risks will require continued development of verification mechanisms, evaluation frameworks, and training objectives that encourage truthful reasoning and discourage unsupported conclusions.

Finally, this research highlights the importance of aligning AI reasoning processes with human problem-solving practices. The structured reasoning paradigms explored in this thesis—such as hypothesis generation, verification, and iterative refinement—reflect cognitive patterns commonly observed in human reasoning. Training models with data that captures these structures may encourage systems that interact more naturally with human analytical workflows. Such alignment may improve collaboration between humans and AI systems, particularly in settings where models are used as assistants for complex reasoning tasks.

Bibliography

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024a. URL <https://arxiv.org/abs/2412.08905>.
- Marah Abdin, Jyoti Aneja, and et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024b. URL <https://arxiv.org/abs/2404.14219>.
- Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.04697>.
- Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1203. URL <https://aclanthology.org/D16-1203>.
- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don’t just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980, 2018.
01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2025a. URL <https://arxiv.org/abs/2403.04652>.
- Essential AI, :, Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Anthony Polloreno, Ashish Tanwer, Burhan Drak Sibai, Divya S Mansingka, Divya Shivaprasad, Ishaan Shah, Karl Stratos, Khoi Nguyen, Michael Callahan, Michael Pust, Mrinal Iyer, Philip Monk, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, and Tim Romanski. Rethinking reflection in pre-training, 2025b. URL <https://arxiv.org/abs/2504.04022>.
- AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

-
- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL <https://aclanthology.org/2023.emnlp-main.298>.
- Syeda Nahida Akter, Zichun Yu, Aashiq Muhamed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. An in-depth look at gemini’s language abilities, 2023.
- Syeda Nahida Akter, Shrimai Prabhumoye, John Kamalu, Sanjeev Satheesh, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Mind: Math informed synthetic dialogues for pretraining llms, 2024. URL <https://arxiv.org/abs/2410.12881>.
- Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-crossthink: Scaling self-learning beyond math reasoning, 2025. URL <https://arxiv.org/abs/2504.13941>.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL <https://arxiv.org/abs/2502.02737>.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 39–48, 2016.
- Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- Alessandro Antonietti. Chapter 15 why does mental visualization facilitate problem-solving? In Robert H. Logie and Michel Denis (eds.), *Mental Images in Human Cognition*, volume 80 of *Advances in Psychology*, pp. 211–227. North-Holland, 1991. doi: [https://doi.org/10.1016/S0166-4115\(08\)60515-4](https://doi.org/10.1016/S0166-4115(08)60515-4). URL <https://www.sciencedirect.com/science/article/pii/S0166411508605154>.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023a.
- Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023b.

-
- Minryoung Bae and Byungmin Lee. Effects of text length and question type on test-takers' performance on fill-in-the-blank items in Korean CSAT. *English Teaching*, 73(4):149–174, 2018.
- Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <https://aclanthology.org/W05-0909>.
- Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 105–121, 2018.
- André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. Comprehensive exploration of synthetic data generation: A survey, 2024. URL <https://arxiv.org/abs/2401.02524>.
- Malcolm I. Bauer and P. N. Johnson-Laird. How diagrams can improve reasoning. *Psychological Science*, 4(6):372–378, 1993. ISSN 09567976, 14679280. URL <http://www.jstor.org/stable/40062565>.
- Annette Baumgaertner, Cornelius Weiller, and Christian Büchel. Event-related fMRI reveals cortical sites involved in contextual sentence integration. *NeuroImage*, 16(3):736–745, 2002.
- Edward Beeching, Shengyi Costa Huang, Albert Jiang, Jia Li, Benjamin Lipkin, Zihan Qina, Kashif Rasul, Ziju Shen, Roman Soletskyi, and Lewis Tunstall. NuminaMath 7B CoT. <https://huggingface.co/AI-MO/NuminaMath-7B-CoT>, 2024.
- Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiu Shi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.
- Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. The ACL Anthology Reference Corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the sixth international conference on language resources and evaluation (LREC'08)*, 2008.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

-
- Robert A Bjork and Marcia C Linn. The science of learning and the learning of science. *Aps Observer*, 19(3), 2006.
- Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training, 2024. URL <https://arxiv.org/abs/2406.03476>.
- A.J.H. Boonen, F. van Wesel, J. Jolles, and M. van der Schoot. The role of visual representation type, spatial ability, and reading comprehension in word problem solving: An item-level analysis in elementary school children. *International Journal of Educational Research*, 68(4): 15–26, 2014. ISSN 0883-0355. doi: 10.1016/j.ijer.2014.08.001.
- Thorsten Brants and Alex Franz. All our n-gram are belong to you. <https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html>, 2006. Accessed: 2023-05-22.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020b.
- Eivinas Butkus and Nikolaus Kriegeskorte. Causal discovery and inference through next-token prediction. In *Advances in Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=MMYTA3v66p>.
- Zheng Cai et al. Internlm2 technical report, 2024. URL <https://arxiv.org/abs/2403.17297>.
- Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas, 2024. URL <https://arxiv.org/abs/2406.20094>.
- Delong Chen, Jianfeng Liu, Wenliang Dai, and Baoyuan Wang. Visual instruction tuning with polite flamingo. *arXiv preprint arXiv:2307.01003*, 2023a.
- Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training rl-like reasoning large vision-language models, 2025a. URL <https://arxiv.org/abs/2504.11468>.
- Jierun Chen, Tiezheng Yu, Haoli Bai, Lewei Yao, Jiannan Wu, Kaican Li, Fei Mi, Chaofan Tao, Lei Zhu, Manyi Zhang, Xiaohui Li, Lu Hou, Lifeng Shang, and Qun Liu. The synergy dilemma of long-cot sft and rl: Investigating post-training techniques for reasoning vlms, 2025b. URL <https://arxiv.org/abs/2507.07562>.

-
- Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023b.
- Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt’s behavior changing over time?, 2023c.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgan Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021a. URL <https://arxiv.org/abs/2107.03374>.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021b.
- Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning, 2025c. URL <https://arxiv.org/abs/2505.16400>.
- Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023d.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.
- Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2529–2550, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.148. URL <https://aclanthology.org/2024.emnlp-main.148/>.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL <https://arxiv.org/abs/2501.17161>.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL <https://arxiv.org/abs/1803.05457>.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021a.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021b. URL <https://arxiv.org/abs/2110.14168>.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021c.

Jeremy R. Cole, Aditi Chaudhary, Bhuwan Dhingra, and Partha Talukdar. Salient span masking for temporal understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3052–3060, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.eacl-main.222>.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

-
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023a.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023b.
- Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Zhao, Aida Amini, Mike Green, Qazi Rashid, and Kelvin Guu. Dialog inpainting: Turning documents to dialogs. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.
- Tri Dao and Albert Gu. Transformers are SSMS: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/dao24a.html>.
- DatologyAI, :, Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, Charvi Bannur, Christina Baek, Darren Teh, David Schwab, Haakon Mongstad, Haoli Yin, Josh Wills, Kaleigh Mentzer, Luke Merrick, Ricardo Monti, Rishabh Adiga, Siddharth Joshi, Spandan Das, Zhengping Wang, Bogdan Gaza, Ari Morcos, and Matthew Leavitt. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining, 2025. URL <https://arxiv.org/abs/2508.10975>.
- Elisa De Stefani and Dorian De Marco. Language, gesture, and emotional communication: An embodied view of social interaction. *Frontiers in Psychology*, 10:2063, 2019.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL <https://arxiv.org/abs/2501.12948>.
- DeepSeek-AI. Deepseek-v3 technical report, 2025b. URL <https://arxiv.org/abs/2412.19437>.
- Stanislas Dehaene. *The number sense: How the mind creates mathematics*. OUP USA, 2011.
- Lucio M. Dery, Paul Michel, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. AANG : Automating auxiliary learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vtVDI3w_BLL.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423/>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019*

-
- conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp. 4171–4186, 2019b.
- Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Jacob Eisenstein. What to do about bad language on the internet. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 359–369, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <https://aclanthology.org/N13-1037/>.
- Ronen Eldan and Yuanzhi Li. Tinstories: How small can language models be and still speak coherent english?, 2023. URL <https://arxiv.org/abs/2305.07759>.
- Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Maximize your data’s potential: Enhancing llm accuracy with two-phase pretraining. *arXiv preprint arXiv:2412.15285*, 2024a.
- Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Maximize your data’s potential: Enhancing llm accuracy with two-phase pretraining, 2024b. URL <https://arxiv.org/abs/2412.15285>.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025a. URL <https://arxiv.org/abs/2503.01307>.
- Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025b. URL <https://openreview.net/forum?id=QGJ9ttXLTy>.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang.

-
- Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024a. URL <https://arxiv.org/abs/2410.07985>.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020a. URL <https://arxiv.org/abs/2101.00027>.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020b.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024b. URL <https://zenodo.org/records/12608602>.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/gao23f.html>.
- Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas, 2024. URL <https://arxiv.org/abs/2406.20094>.
- Gemini. Gemini: A family of highly capable multimodal models, 2024. URL <https://arxiv.org/abs/2312.11805>.
- Gael Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not strong abstract reasoners yet. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024. URL <https://openreview.net/forum?id=Pc0fPGip78>.
- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järvinemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2024. URL <https://arxiv.org/abs/2411.04872>.
- Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model, 2024. URL <https://arxiv.org/abs/2405.16712>.
- GoogleAI. Bard, 2023.

-
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. Train no evil: Selective masking for task-guided pre-training. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6966–6974, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.566. URL <https://aclanthology.org/2020.emnlp-main.566/>.
- Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanxia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL <https://arxiv.org/abs/2506.04178>.
- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL <https://arxiv.org/abs/2306.11644>.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024a. URL <https://arxiv.org/abs/2401.14196>.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Yiduo Guo, Jie Fu, Huishuai Zhang, Dongyan Zhao, and Yikang Shen. Efficient continual pre-training by mitigating the stability gap. *arXiv preprint arXiv:2406.14833*, 2024b.
- Sonam Gupta, Yatin Nandwani, Asaf Yehudai, Dinesh Khandelwal, Dinesh Raghu, and Sachindra Joshi. Selective self-to-supervised fine-tuning for generalization in large language models, 2025. URL <https://arxiv.org/abs/2502.08130>.
- Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, June 2023.

-
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, 2020.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pp. 3929–3938. PMLR, 2020.
- Peter Hagoort, Lea Hald, Marcel Bastiaansen, and Karl Magnus Petersson. Integration of word meaning and world knowledge in language comprehension. *science*, 304(5669): 438–441, 2004.
- Seungwook Han, Jyothish Pari, Samuel J Gershman, and Pulkit Agrawal. Position: General intelligence requires reward-based pretraining. In *Forty-second International Conference on Machine Learning Position Paper Track*.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL <https://aclanthology.org/2023.emnlp-main.507>.
- Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL <https://arxiv.org/abs/2009.03300>.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021c. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021d.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021e.

-
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022a.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf.
- Joy Hsu, Gabriel Poesia, Jiajun Wu, and Noah D. Goodman. Can visual scratchpads with diagrammatic abstractions augment LLM reasoning? In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*, 2023. URL <https://openreview.net/forum?id=YlhKbQ0zF3>.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. <https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero>, 2025a.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025b. URL <https://arxiv.org/abs/2503.24290>.
- Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning, 2025. URL <https://arxiv.org/abs/2507.00432>.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

-
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024a. URL <https://arxiv.org/abs/2310.01798>.
- Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. Lawyer llama technical report. *arXiv preprint arXiv:2305.15062*, 2023.
- Yiming Huang, Xiao Liu, Yeyun Gong, Zhibin Gou, Yelong Shen, Nan Duan, and Weizhu Chen. Key-point-driven data synthesis with its enhancement on mathematical reasoning, 2024b. URL <https://arxiv.org/abs/2403.02333>.
- Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey—part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024c.
- Zhongzhen Huang, Gui Geng, Shengyi Hua, Zhen Huang, Haoyang Zou, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. O1 replication journey – part 3: Inference-time scaling for medical reasoning. *arXiv preprint arXiv:2501.06458*, 2025.
- Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019a.
- Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*, 2018.
- Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019b.
- Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL <https://openreview.net/forum?id=DimPeeCxKO>.
- Yoichi Ishibashi, Taro Yano, and Masafumi Oyamada. Mining hidden thoughts from texts: Evaluating continual pretraining with synthetic data for llm reasoning, 2025. URL <https://arxiv.org/abs/2505.10182>.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=chfJYC3iL>.
- Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL <https://aclanthology.org/N19-1357>.

-
- Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1:3, 2023.
- Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL <https://aclanthology.org/D17-1215>.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL <https://arxiv.org/abs/2310.06825>.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL <https://arxiv.org/abs/2401.04088>.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=VTF8yNQM66>.
- Jiahe Jin, Abhijay Paladugu, and Chenyan Xiong. Beneficial reasoning behaviors in agentic search and effective post-training to obtain them, 2026. URL <https://arxiv.org/abs/2510.06534>.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.
- Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.
- Armand Joulin. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.
- David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406, 2018.
- Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017.

-
- Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011. ISBN 9780374275631 0374275637. URL https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDL7.
- Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzas, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *Computer Vision – ECCV 2022*, pp. 300–318. Springer Nature Switzerland, 2022. ISBN 978-3-031-20056-4. doi: 10.1007/978-3-031-20056-4_18. URL https://link.springer.com/chapter/10.1007/978-3-031-20056-4_18.
- Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls, 2025. URL <https://arxiv.org/abs/2510.01631>.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Zixuan Ke, Yijia Shao, Haowei Lin, Hu Xu, Lei Shu, and Bing Liu. Adapting a language model while preserving its general knowledge. *arXiv preprint arXiv:2301.08986*, 2023.
- Johannes Keck, Adam Zabicki, Julia Bachmann, Jörn Munzert, and Britta Krüger. Decoding spatiotemporal features of emotional body language in social interactions. *Scientific Reports*, 12(1):15088, 2022.
- Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL <https://aclanthology.org/N16-1136>.
- Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17, 2015.
- Jennifer L. Krawec. Problem representation and mathematical problem solving of students of varying math ability. *Journal of Learning Disabilities*, 47(2):103–115, 2014. doi: 10.1177/0022219412436976. URL <https://doi.org/10.1177/0022219412436976>. PMID: 22392891.
- Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureaux. Chemprot-3.0: a global chemical biology diseases mapping. *Database*, 2016, 2016.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL <https://arxiv.org/abs/1602.07332>.

-
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL <https://aclanthology.org/D17-1082>.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- Teven Le Scao, Thomas Wang, Daniel Hesslow, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, Ofir Press, Colin Raffel, Victor Sanh, Sheng Shen, Lintang Sutawika, Jaesung Tae, Zheng Xin Yong, Julien Launay, and Iz Beltagy. What language model to train if you have one million GPU hours? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 765–782, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.54. URL <https://aclanthology.org/2022.findings-emnlp.54/>.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL <https://doi.org/10.1093/bioinformatics/btz682>.
- Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pp. 18893–18912. PMLR, 2023.
- Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, compositional video question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1369–1379, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1167. URL <https://aclanthology.org/D18-1167/>.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022a. URL <https://arxiv.org/abs/2206.14858>.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022b.
- Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities, 2024a. URL <https://arxiv.org/abs/2403.04706>.

-
- Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. Culturellm: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838, 2024b.
- Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*, 2024c.
- Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruva Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024d.
- Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wieggers, and Zhiyong Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016, 05 2016. ISSN 1758-0463. doi: 10.1093/database/baw068. URL <https://doi.org/10.1093/database/baw068>. baw068.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023a.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023b. URL <https://arxiv.org/abs/2309.05463>.
- Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021.
- Yiming Liang, Tianyu Zheng, Xinrun Du, Ge Zhang, Jiaheng Liu, Xingwei Qu, Wenqiang Zu, Xingrun Xing, Chujie Zheng, Lei Ma, Guoyin Wang, Zhaoxiang Zhang, Wenhao Huang, Xiang Yue, and Jiajun Zhang. Aligning instruction tuning with pre-training, 2025. URL <https://arxiv.org/abs/2501.09368>.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. EntityBERT: Entity-centric masking strategy for model pretraining for the clinical domain. In Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii (eds.), *Proceedings of the 20th Workshop on Biomedical Language Processing*, pp. 191–201, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.bionlp-1.21. URL <https://aclanthology.org/2021.bionlp-1.21/>.

-
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL <https://aclanthology.org/2022.acl-long.229>.
- Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. Tinygsm: achieving >80 URL <https://arxiv.org/abs/2312.09241>.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023b.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572. Curran Associates, Inc., 2023c. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/43e9d647ccd3e4b7b5baab53f0368686-Paper-Conference.pdf.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025a. URL <https://arxiv.org/abs/2505.24864>.
- Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data, 2024b. URL <https://arxiv.org/abs/2404.07503>.
- Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. Llava-plus: Learning to use tools for creating multimodal agents, 2023d.
- Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. Nover: Incentive training for language models via verifier-free reinforcement learning. *arXiv preprint arXiv:2505.16022*, 2025b.
- Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy, 2025c. URL <https://arxiv.org/abs/2506.13284>.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In

-
- Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11065–11082, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.658. URL <https://aclanthology.org/2024.findings-acl.658/>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024. URL <https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu>.
- Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *The 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaler: Surpassing o1-preview with a 1.5b model by scaling rl. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>, 2025a. Notion Blog.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025b. URL <https://arxiv.org/abs/2308.08747>.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- Pratyush Maini. Phi-1.5 model: A case of comparing apples to oranges? 2023. URL https://pratyushmaini.github.io/phi-1_5/.
- Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Data Problems for Foundation Models Workshop at ICLR*, 2024a. URL <https://arxiv.org/abs/2401.16380>.
- Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In

-
- Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14044–14072, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.757. URL <https://aclanthology.org/2024.acl-long.757/>.
- Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477, 2022.
- Paul Metzner, Titus von der Malsburg, Shravan Vasishth, and Frank Rösler. Brain responses to world knowledge violations: A comparison of stimulus-and fixation-triggered event-related potentials and neural oscillations. *Journal of Cognitive Neuroscience*, 27(5):1017–1028, 2015.
- Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL <https://aclanthology.org/2020.acl-main.92>.
- Microsoft Research. Phi-2: The surprising power of small language models. <https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/>, 2023. Accessed: 2025-03-28.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.
- George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL <https://aclanthology.org/H94-1111>.
- Philipp Mondorf and Barbara Plank. Beyond accuracy: Evaluating the reasoning behavior of large language models – a survey, 2024. URL <https://arxiv.org/abs/2404.01869>.
- Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset, 2025. URL <https://arxiv.org/abs/2504.16891>.
- Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models, 2023. URL <https://arxiv.org/abs/2305.16264>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.
- NVIDIA Nemotron Nano. Efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025.

-
- Binh X Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Coarse-to-fine reasoning for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4566, 2022.
- Peter Norvig. Natural language corpus data. *Beautiful data*, pp. 219–242, 2009.
- NVIDIA. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models, 2025a. URL <https://arxiv.org/abs/2504.03624>.
- NVIDIA. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model, 2025b. URL <https://arxiv.org/abs/2508.14444>.
- Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzec, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhunoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report, 2024. URL <https://arxiv.org/abs/2406.11704>.
- NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwala, Andrew Tao, Anjulie Agrusa, Ankur Verma, Ann Guan, Anubhav Mandarwal, Arham Mehta, Ashwath Aithal, Ashwin Poojary, Asif Ahamed, Asit Mishra, Asma Kuriparambil Thekkumpate, Ayush Dattagupta, Banghua Zhu, Bardiya Sadeghi, Barnaby Simkin, Ben Lanir, Benedikt Schifferer, Besmira Nushi, Bilal Kartal, Bitu Darvish Rouhani, Boris Ginsburg, Brandon Norrick, Brandon Soubasis, Branislav Kisacanin, Brian Yu, Bryan Catanzaro, Carlo del Mundo, Chantal Hwang, Charles Wang, Cheng-Ping Hsieh, Chenghao Zhang, Chenhan Yu, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christopher Parisien, Collin Neale, Cyril Meurillon, Damon Mosk-Aoyama, Dan Su, Dane Corneil, Daniel Afrimi, Daniel Lo, Daniel Rohrer, Daniel Serebrenik, Daria Gitman, Daria Levy, Darko Stosic, David Mosallanezhad, Deepak Narayanan, Dhruv Nathawani, Dima Rekish, Dina Yared, Divyanshu Kakwani, Dong Ahn, Duncan Riach, Dusan Stosic, Edgar Minasyan, Edward Lin, Eileen Long, Eileen Peters Long, Elad Segal, Elena Lantz, Ellie Evans, Elliott Ning, Eric Chung, Eric Harper, Eric Tramel, Erick Galinkin, Erik Pounds, Evan Briones, Evelina Bakhturina, Evgeny Tsykunov, Faisal Ladhak, Fay Wang, Fei Jia, Felipe Soares,

Feng Chen, Ferenc Galko, Frank Sun, Frankie Siino, Gal Hubara Agam, Ganesh Ajjanagadde, Gantavya Bhatt, Gargi Prasad, George Armstrong, Gerald Shen, Gorkem Batmaz, Grigor Nalbandyan, Haifeng Qian, Harsh Sharma, Hayley Ross, Helen Ngo, Herbert Hum, Herman Sahota, Hexin Wang, Himanshu Soni, Hiren Upadhyay, Huizi Mao, Huy C Nguyen, Huy Q Nguyen, Iain Cunningham, Ido Galil, Ido Shahaf, Igor Gitman, Ilya Loshchilov, Itamar Schen, Itay Levy, Ivan Moshkov, Izik Golan, Izzy Putterman, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jatin Mitra, Jeffrey Glick, Jenny Chen, Jesse Oliver, Jian Zhang, Jiaqi Zeng, Jie Lou, Jimmy Zhang, Jinhang Choi, Jining Huang, Joey Conway, Joey Guman, John Kamalu, Johnny Greco, Jonathan Cohen, Joseph Jennings, Joyjit Daw, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kai Xu, Kan Zhu, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kevin Shih, Kezhi Kong, Khushi Bhardwaj, Kirthi Shankar, Krishna C. Puvvada, Krzysztof Pawelec, Kumar Anik, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Li Ding, Lizzie Wei, Lucas Liebenwein, Luis Vega, Maanu Grover, Maarten Van Segbroeck, Maer Rodrigues de Melo, Mahdi Nazemi, Makesh Narsimhan Sreedhar, Manoj Kilaru, Maor Ashkenazi, Marc Romeijn, Marcin Chochowski, Mark Cai, Markus Kliegl, Maryam Moosaei, Matt Kulka, Matvei Novikov, Mehrzad Samadi, Melissa Corpuz, Mengru Wang, Meredith Price, Michael Andersch, Michael Boone, Michael Evans, Miguel Martinez, Mikail Khona, Mike Chrzanowski, Minseok Lee, Mohammad Dabbah, Mohammad Shoeybi, Mostofa Patwary, Nabin Mulepati, Najeeb Nabwani, Natalie Hereth, Nave Assaf, Negar Habibi, Neta Zmora, Netanel Haber, Nicola Sessions, Nidhi Bhatia, Nikhil Jukar, Nikki Pope, Nikolai Ludwig, Nima Tajbakhsh, Nir Ailon, Nirmal Juluru, Nishant Sharma, Oleksii Hrinchuk, Oleksii Kuchaiev, Olivier Delalleau, Oluwatobi Olabiyi, Omer Ullman Argov, Omri Puny, Oren Tropp, Ouye Xie, Parth Chadha, Pasha Shamis, Paul Gibbons, Pavlo Molchanov, Pawel Morkisz, Peter Dykas, Peter Jin, Pinky Xu, Piotr Januszewski, Pranav Prashant Thombre, Prason Varshney, Pritam Gundecha, Przemek Tredak, Qing Miao, Qiyu Wan, Rabeeh Karimi Mahabadi, Rachit Garg, Ran El-Yaniv, Ran Zilberstein, Rasoul Shafipour, Rich Harang, Rick Izzo, Rima Shahbazyan, Rishabh Garg, Ritika Borkar, Ritu Gala, Riyad Islam, Robert Hesse, Roger Waleffe, Rohit Watve, Roi Koren, Ruoxi Zhang, Russell Hewett, Russell J. Hewett, Ryan Prenger, Ryan Timbrook, Sadegh Mahdavi, Sahil Modi, Samuel Krیمان, Sangkug Lim, Sanjay Kariyappa, Sanjeev Satheesh, Saori Kaji, Satish Pasumarthi, Saurav Muralidharan, Sean Narentharen, Sean Narenthiran, Seonmyeong Bak, Sergey Kashirsky, Seth Poulos, Shahar Mor, Shanmugam Ramasamy, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shiqing Fan, Shreya Gopal, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Simeng Sun, Smita Ithape, Somshubra Majumdar, Soumye Singhal, Stas Sergienko, Stefania Alborghetti, Stephen Ge, Sugam Dipak Devare, Sumeet Kumar Barua, Suseella Panguluri, Suyog Gupta, Sweta Priyadarshi, Syeda Nahida Akter, Tan Bui, Teodor-Dumitru Ene, Terry Kong, Thanh Do, Tijmen Blankevoort, Tim Moon, Tom Balough, Tomer Asida, Tomer Bar Natan, Tomer Ronen, Tugrul Konuk, Twinkle Vashishth, Udi Karpas, Ushnish De, Vahid Noorozi, Vahid Noroozi, Venkat Srinivasan, Venmugil Elango, Victor Cui, Vijay Korthikanti, Vinay Rao, Vitaly Kurin, Vitaly Lavruchkin, Vladimir Anisimov, Wanli Jiang, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenfei Zhou, Will Jennings, William Zhang, Wojciech Prazuch, Xiaowei Ren, Yashaswi Karnati, Yejin Choi, Yev Meyer, Yi-Fu Wu, Yian Zhang, Yigong Qin, Ying Lin, Yonatan Geifman, Yonggan Fu, Yoshi Subara, Yoshi Suhara, Yubo Gao, Zach Moshe, Zhen Dong, Zhongbo Zhu, Zihan

-
- Liu, Zijia Chen, and Zijie Yan. Nvidia nemotron 3: Efficient and open intelligence, 2025. URL <https://arxiv.org/abs/2512.20856>.
- Augustus Odena, Charles Sutton, David Martin Dohan, Ellen Jiang, Henryk Michalewski, Jacob Austin, Maarten Paul Bosma, Maxwell Nye, Michael Terry, and Quoc V. Le. Program synthesis with large language models. In *n/a*, pp. n/a, n/a, 2021. n/a.
- Team OLMo. 2 olmo 2 furious, 2025. URL <https://arxiv.org/abs/2501.00656>.
- OpenAI. Openai gpt-3.5 api [gpt-3.5-turbo], 2021.
- OpenAI. Gpt-4 technical report, 2023.
- OpenAI. Gpt-4o (openai o1), 2024a. URL <https://openai.com/o1/>.
- OpenAI. Gpt-4 technical report, 2024b. URL <https://arxiv.org/abs/2303.08774>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Frank Pajares and M David Miller. Mathematics self-efficacy and mathematical problem solving: Implications of using different forms of assessment. *The journal of experimental education*, 65(3):213–228, 1997.
- Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Data, data everywhere: A guide for pretraining dataset construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10671–10695, 2024a.
- Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Dan Su, Chen Zhu, Deepak Narayanan, Aastha Jhunjhunwala, Ayush Dattagupta, et al. Nemotron-4 15b technical report. *arXiv preprint arXiv:2402.16819*, 2024b.
- Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don’t retrain: A recipe for continued pretraining of language models, 2024c. URL <https://arxiv.org/abs/2407.07263>.
- Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.
- Ajay Patel, Colin Raffel, and Chris Callison-Burch. DataDreamer: A tool for synthetic data generation and reproducible LLM workflows. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3781–3799, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.acl-long.208>.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL <https://aclanthology.org/2021.naacl-main.168>.

-
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL <https://arxiv.org/abs/2406.17557>.
- Long Phan et al. Humanity’s last exam, 2025. URL <https://arxiv.org/abs/2501.14249>.
- Gabriel Poesia, Kanishk Gandhi, Eric Zelikman, and Noah D. Goodman. Certified reasoning with language models. 2023.
- Vignesh Prabhakar, Md Amirul Islam, Adam Atanas, Yao-Ting Wang, Joah Han, Aastha Jhunjhunwala, Rucha Apte, Robert Clark, Kang Xu, Zihan Wang, and Kai Liu. Omniscience: A domain-specialized llm for scientific reasoning and discovery, 2025. URL <https://arxiv.org/abs/2503.17604>.
- Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report—part 1. *arXiv preprint arXiv:2410.18982*, 2024.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021a. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021b.
- Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020a. URL <http://jmlr.org/papers/v21/20-074.html>.

-
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020b.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL <https://aclanthology.org/P18-2124>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=Ti67584b98>.
- Paul M. Riechers, Henry R. Bigelow, Eric A. Alt, and Adam Shai. Next-token pretraining implies in-context learning, 2025. URL <https://arxiv.org/abs/2505.18373>.
- Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI’17*, pp. 2662–2670. AAAI Press, 2017. ISBN 9780999241103.
- Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in Neural Information Processing Systems*, 35:9512–9524, 2022.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL <https://aclanthology.org/D19-1454>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. Squinting at vqa models: Introspecting vqa models with sub-questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10003–10011, 2020.

-
- Amrith Setlur, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms, 2025. URL <https://arxiv.org/abs/2506.09026>.
- Vedant Shah, Dingli Yu, Kaifeng Lyu, Simon Park, Nan Rosemary Ke, Michael Mozer, Yoshua Bengio, Sanjeev Arora, and Anirudh Goyal. Ai-assisted generation of difficult math questions, 2024. URL <https://arxiv.org/abs/2407.21009>.
- Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Noam Shazeer. Glu variants improve transformer, 2020. URL <https://arxiv.org/abs/2002.05202>.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.
- Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions, 2020.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019a.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019b.
- Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhunoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.
- Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxin Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024. URL <https://arxiv.org/abs/2402.00159>.
- Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

-
- Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2017. URL <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972>.
- Simon St.Laurent, Murata Makoto, and Dan Kohn. XML Media Types. RFC 3023, January 2001. URL <https://www.rfc-editor.org/info/rfc3023>.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3645–3650. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1355. URL <https://doi.org/10.18653/v1/p19-1355>.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.
- Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache, 2019.
- Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5594–5608, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.495. URL <https://aclanthology.org/2020.acl-main.495>.
- Sanjay Subramanian, Medhini Narasimhan, Kushal Khangaonkar, Kevin Yang, Arsha Nagrani, Cordelia Schmid, Andy Zeng, Trevor Darrell, and Dan Klein. Modular visual question answering via code generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, 2023. Association for Computational Linguistics.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6418–6428, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1644. URL <https://aclanthology.org/P19-1644>.
- Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American*

-
- Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL <https://aclanthology.org/N19-1421>.
- Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- Till Tantau. *The TikZ and PGF Packages*, April 2022. URL <https://github.com/pgf-tikz/pgf>. Manual for version 3.1.9a.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- Gemini Team. Gemini: A family of highly capable multimodal models, 2023.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimentko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL <https://arxiv.org/abs/2403.08295>.
- Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025a. URL <https://arxiv.org/abs/2501.12599>.
- M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding,

-
- Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Tianyang Pang, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL <https://arxiv.org/abs/2502.14739>.
- Qwen Team. Qwen2.5: A party of foundation models, September 2024a. URL <https://qwenlm.github.io/blog/qwen2.5/>.
- Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024b. URL <https://qwenlm.github.io/blog/qwq-32b-preview/>.
- Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025b. URL <https://qwenlm.github.io/blog/qwq-32b/>.
- Josh Tenenbaum. Building machines that learn and think like people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18*, pp. 5, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.
- Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanic, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024a.
- Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset, 2024b. URL <https://arxiv.org/abs/2402.10176>.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023a.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023b. URL <https://arxiv.org/abs/2302.13971>.
- Trieu Trinh, Yuhuai Tony Wu, Quoc Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024. URL <https://www.nature.com/articles/s41586-023-06747-5>.
- Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. ReFT: Reasoning with reinforced fine-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7601–7614, Bangkok, Thailand, August 2024.

-
- Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.410. URL <https://aclanthology.org/2024.acl-long.410/>.
- Delinda van Garderen, Amy Scheuermann, and Christa Jackson. Developing representational ability in mathematics for students with learning disabilities: A content analysis of grades 6 and 7 textbooks. *Learning Disability Quarterly*, 35(1):24–38, 2012. doi: 10.1177/0731948711429726. URL <https://doi.org/10.1177/0731948711429726>.
- Delinda van Garderen, Amy Scheuermann, Apryl Poch, and Mary M. Murray. Visual representation in mathematics: Special education teachers’ knowledge and emphasis for instruction. *Teacher Education and Special Education*, 41(1):7–23, 2018. doi: 10.1177/0888406416665448. URL <https://doi.org/10.1177/0888406416665448>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017b.
- Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL <https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024>.
- Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data, 2024. URL <https://arxiv.org/abs/2211.04325>.
- Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization, 2024a. URL <https://arxiv.org/abs/2405.15071>.
- Taiyi Wang, Sian Gooding, Florian Hartmann, Oriana Riva, and Edward Grefenstette. A subgoal-driven framework for improving long-horizon llm agents, 2026a. URL <https://arxiv.org/abs/2603.19685>.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023a. URL <https://openreview.net/forum?id=1PL1NIMMrw>.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang,

-
- Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhua Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhua Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024c. URL <https://arxiv.org/abs/2406.01574>.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024d.
- Zehong Wang, Fang Wu, Hongru Wang, Xiangru Tang, Bolian Li, Zhenfei Yin, Yijun Ma, Yiyang Li, Weixiang Sun, Xiushi Chen, and Yanfang Ye. Why reasoning fails to plan: A planning-centric analysis of long-horizon decision making in llm agents, 2026b. URL <https://arxiv.org/abs/2601.22311>.
- Zengzhi Wang, Rui Xia, and Pengfei Liu. Generative ai for math: Part i – mathpile: A billion-token-scale pretraining corpus for math. *arXiv preprint arXiv:2312.17120*, 2023b.
- Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025. Preprint.
- Zhongdao Wang, Hengshuang Zhao, Ya-Li Li, Shengjin Wang, Philip Torr, and Luca Bertinetto. Do different tracking tasks require different appearance models? *Advances in Neural Information Processing Systems*, 34:726–738, 2021.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
- Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, et al. Agentic reasoning for large language models. *arXiv preprint arXiv:2601.12538*, 2026.
- Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*, 2021.
- Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng

Zhang. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond, 2025. URL <https://arxiv.org/abs/2503.10460>.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. QuRating: Selecting high-quality data for training language models. In *International Conference on Machine Learning (ICML)*, 2024.

Alex Wilf, Martin Q. Ma, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Face-to-face contrastive learning for social intelligence question-answering. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–7, 2023. doi: 10.1109/FG57933.2023.10042612.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rhea Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi,

Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwā, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynek, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Perrián, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrmann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sängler, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aaronsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL <https://arxiv.org/abs/2211.05100>.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning, 2025a. URL <https://arxiv.org/abs/2410.23123>.

-
- Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, 2025b. URL <https://arxiv.org/abs/2502.14768>.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.
- Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling LLM under token-crisis. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=Af5GvIj3T5>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.
- Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Neural Information Processing Systems (NeurIPS)*, 2023.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.
- Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. Synthetic continued pretraining. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 44379–44421, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.
- Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*, 2022.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl: Modularization empowers large language models with multimodality, 2023.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL <https://arxiv.org/abs/2502.03387>.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms, 2025. URL <https://arxiv.org/abs/2502.03373>.
- Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

-
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=N8N0hgNDRt>.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3, 2024.
- Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. Naturalreasoning: Reasoning in the wild with 2.8m challenging questions, 2025. URL <https://arxiv.org/abs/2502.13124>.
- Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhua Chen. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*, 2024.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL <https://arxiv.org/abs/2504.13837>.
- Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8807–8817, 2019.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16375–16387, 2022.
- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL <https://arxiv.org/abs/2503.18892>.
- Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations (ICLR)*, 2024.
- Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016a.
- Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b.

-
- Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5014–5022, 2016c.
- Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well?, 2025a. URL <https://arxiv.org/abs/2503.24235>.
- Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest, 2023a.
- Zeyu Zhang et al. Measuring in-context computation complexity via hidden state prediction, 2025b. URL <https://arxiv.org/abs/2503.13431>.
- Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multi-modal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023b.
- Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multi-modal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023c.
- Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*, 2024.
- Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.
- Qihao Zhu Runxin Xu Junxiao Song Mingchuan Zhang Y.K. Li Y. Wu Daya Guo Zhihong Shao, Peiyi Wang. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.
- Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. Megamath: Pushing the limits of open math corpora, 2025a. URL <https://arxiv.org/abs/2504.02807>.
- Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*, 2025b.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023a.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Graph-informed dynamic evaluation of large language models. *arXiv preprint arXiv:2309.17167*, 2023b.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dynamic evaluation of large language models by meta probing agents. *arXiv preprint arXiv:2402.14865*, 2024.

Chapter 12

Appendix for Anatomy of Data

This appendix provides supplementary material for [Part I](#), “Anatomy of High Quality Data for Reasoning.” It is organized by the four data-synthesis threads covered in the main text. **VisReas** ([chapter 3](#)): data balancing and clustering strategy; an overview and analysis of the VisReas dataset (question types, relation hops, readability, pseudocode format); baseline configuration; the effect of pseudocode fine-tuning; example questions from VisReas and GQA; and Mechanical Turk annotation details. **Self-Imagine** ([chapter 4](#)): image-generation and evaluation prompts; prompts used for reasoning and mathematics tasks (question-only vs. question+image); and additional results on accuracy by question length and by chain-of-thought length across GSM8K, SVAMP, ASDIV, and BIG-Bench Hard. **MIND** ([chapter 5](#)): full prompts for each conversational style (Two Professors, Teacher–Student, Two Students, Interview, Problem Solving, Layman Know-All, Debate); evaluation metric and benchmark details; additional experiments and ablations (role-based conversations, pretraining from scratch, context length vs. quality, dialogue examples). **Nemotron-CrossThink** ([chapter 6](#)): data proportion across blending strategies; token-efficiency analysis for correct responses; and sub-category accuracy analysis.

12.1 Data Balancing

A primary concern with current VQA datasets is the prevalence of question-conditional biases, enabling models to make informed guesses without a genuine grasp of the underlying images. Nevertheless, precise rendering of question semantics could offer enhanced control over these biases, holding the potential to significantly mitigate the issue [Zhang et al. \(2016c\)](#); [Kafle & Kanan \(2017\)](#). Motivated by this observation, we perform a rigorous balancing based on question categories, attribute/relation types, and answer distribution.

Adopting the balancing approach outlined in previous research [Hudson & Manning \(2019b\)](#), we employ a clustering strategy based on a fusion of two labels: `<attr/rel_type>` and `<res_type>`. The former denotes attributes or relation names (e.g., *red* or *right*), while the latter signifies reasoning types (e.g., *verify.rel*). We refine the question set within each cluster, filtering out questions that encompass overlapping sets of objects in their texts or that contain subsets of objects already covered by other questions with complete sets. We prioritize questions featuring larger sets of objects and multihop relations, provided their length stays below 25. Finally, we introduce an additional label `<answer>` and equilibrate

the question sampling through the answer distribution. After executing this balancing in an iterative manner on 2.07M questions, we generate a balanced corpus of 72,244 questions with images.

12.2 Overview and Analysis of the VISREAS

This section provides an in-depth examination of the VISREAS dataset, focusing on various aspects of question types and their characteristics. It encompasses an overview of question types, the distribution of semantic lengths, question readability scores, average question lengths per reasoning type, the relationship between question frequency and the number of attributes, and human accuracy on attributed questions.

Questions Types and Templates. The VISREAS dataset features a diverse array of question types that challenge multimodal reasoning and compositional understanding. These question types include query, count, compare, verify, and choose, each requiring a unique approach to answer. Depending on how the clusters are made, each question type can further be broken down into attr and rel subtypes. Therefore, in total, there can be nine categories of questions. Figure 12.1 gathers all templates and examples from the dataset to offer insights into the intricacies of these question categories.

Type	Question	Answer	List	No Answer	Example
<query_attr>	a. What <attr> do the <objs> have common?	a. <attr attr>	a. yes	a. yes	What material do the pole and the bike next to the road have common?
	b. What [is are] the <attr> of the <objs>? (can be more specific depending on the attribute type)	b. <attr attr>	b. yes	b. yes	
<count_attr>	a. Among <objs>, how many of them have [a particular <attr> multiple <attrs> a particular <attr> common and one of the object has another <attr>]?	a. <num>	a. no	a. no	Among the shirt, the pant and the hat, how many are red?
	b. [Is Are] there [less than greater than] <num> objects that share [a particular <attr> multiple <attrs>]?	b. <yes no>	b. no	b. no	Are there less than three objects that share red color?
	c. [Is Are] any <num> of the following things, <objs> <attr opp_attr>?	c. <yes no>	c. no	c. no	
<compare_attr>	a. [Does a particular attribute Do multiple attributes] of <obj objs> match with the <attr attr> of <obj objs>?	a. <yes no>	a. no	a. yes	Does the shape of the plate on the table match with the shape of the steel pan and the orange box?
	b. Do <objs> share the same <attr> in the image?	b. <yes no>	b. no	b. yes	
	c. Are <objs> similar in <attr>?	c. <yes no>	c. no	c. yes	
<verify_attr>	Do you see any [<obj objs> of <attr> <attr> <obj objs>] in the image?	<yes no>	no	no	Do you see any red bike and red helmet?
<query_obj>	[What [object objects] Who] in the image [has the same is doing the same] <attr> as <obj objs>?	<obj objs>	yes	yes	Who in the image is the same activity as the boy wearing blue jeans?
<choose_attr>	[Is Are Do Does] <obj objs> [look appear] <attr> or <opp_attr>?	<attr>	no	yes	Do the driver and the passenger look younger or older?
<query_rel>	On which side of the <attr> <obj objs> in the image the <attr> <obj objs> are located?	<rel>	no	yes	On which side of the red car in the image the trees, the metal pole and the silver wire located?
<verify_rel>	[Is Are] the <attr> <obj objs> <rel> <attr attr> <obj objs>?	<yes no>	no	yes	Are the man on the road and the boy with red hair wearing a blue jacket?
<choose_rel>	[Is Are] the <attr> <obj> [located positioned] <rel> <attr attr> <obj objs> or <opp_rel>?	<rel>	no	yes	Are the buses located to the left of the road or to the right of the road?

Figure 12.1: Overview of types of questions along with some templates and examples from the VISREAS corpus.

Distribution of Relation Hops and Readability. A comprehensive analysis of the distribution of relation hops in VISREAS questions reveals a predominant trend toward questions that involve about two reasoning hops. These hops can entail tracking object relations, identifying attributes, or executing logical operations. We conduct a readability test using the workers from Amazon Mechanical Turk. Our analysis reveals that questions with larger relation hops demonstrate a noticeable decline in readability, emphasizing the complexity associated with extended reasoning (Figure 12.2). To enhance the quality of the dataset so that it can reflect the real-world day-to-day life questions, we choose to keep the relation hop within two.

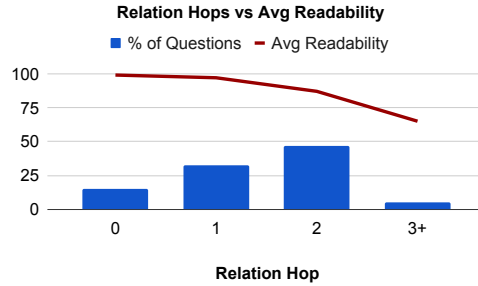
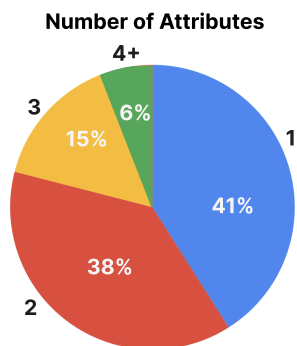


Figure 12.2: Distribution of VISREAS questions semantic length (number of computation steps to arrive at the answer) as well as the readability scores for each semantic step type. We can see that most questions require at most two reasoning steps, where each step may involve tracking a relation between objects, an attribute identification, or a logical operation. At the same time, questions with larger semantic steps are difficult to read.

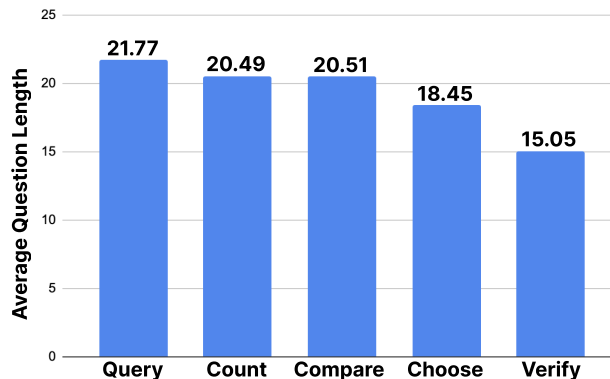
Average Question Length per Reasoning Type. By dissecting question lengths across different reasoning categories in Figure 12.3b, we observe a consistent trend: query questions tend to be longer than other reasoning types. This phenomenon is particularly apparent due to the inclusion of multiple objects sharing similar attributes and their corresponding relations.

Question Frequency and Attribute Usage. The VISREAS corpus has been generated using the clusters of objects that share similar relation or attribute. However, clusters based on shared attributes/relations can share objects that possess all of those attributes/relations. For example, a table and a chair have the color *brown* and material *wood* in an image. Initially, we have two clusters with *brown* and *wood*. Now, if both clusters share some objects, we again create a new cluster based on *brown+wood* adding the shared objects (i.e., table and chair). Using this approach, we create clusters that share multiple attributes and relations and generate questions that involve filtering multiple attributes/relations along with the identification of objects of interest. Figure 12.3a shows the distribution of questions in VISREAS with respect to the number of attributes/relations. As the number of attributes/relations goes higher, the number of clusters also decreases resulting in decreasing number of questions.

Human Accuracy on Attributed Questions. The final facet of our exploration delves into human accuracy when answering attributed questions from the VISREAS dataset. By assessing the performance of human subjects across different question types and attributes,



(a)



(b)

Figure 12.3: (a) Question distribution across the number of attributes in a query. The question complexity increases with the number of attributes or relations. (b) Average question length per reasoning type in VISREAS corpus. Query questions are lengthier than other reasoning categories as these questions contain multiple objects of similar attributes with their relations.

GQA semantic string format

What are the children on?

```
select : children →
relate : on, object, children →
query : name
```

Where in this photo are the green chairs, top or bottom?

```
select : chairs →
filter color : green →
choose vertical position : top | bottom
```

VisReas Pseudocode format

What are the children on?

```
selected_children = select(children)
related_object = relate(_, on, o, selected_children)
result = query(name, related_object)
```

Where in this photo are the green chairs, top or bottom?

```
selected_chairs = select(chairs)
filtered_green = filter_color(green, selected_chairs)
result = choose_vposition(top|bottom, filtered_green)
```

Figure 12.4: **Pseudocode format.** Our method re-structures the format of GQA semantic string to pseudocode to better leverage Code-LLMs without adding any auxiliary information.

we gain a deeper understanding of the challenges inherent to this multimodal reasoning task. Figure 12.6 breaks down the human accuracy across different attribute types. It is noticeable that color and material questions have the lowest accuracy, as they contain a higher amount of questions compared to other attributes.

In summary, this section offers a comprehensive overview and analysis of the VISREAS dataset, encompassing question types, semantic lengths, question readability, average lengths per reasoning type, attribute-based question distribution, and human accuracy. These insights contribute to a holistic understanding of the dataset’s intricacies and its potential to advance the field of visual reasoning and question answering.

12.3 Baseline Configuration

All baselines follow default settings provided by the original author evaluation script. All configurations for model, optimizer, scheduler, and training follow default parameters from Pytorch and Huggingface library. For generative models, all inference is done using default

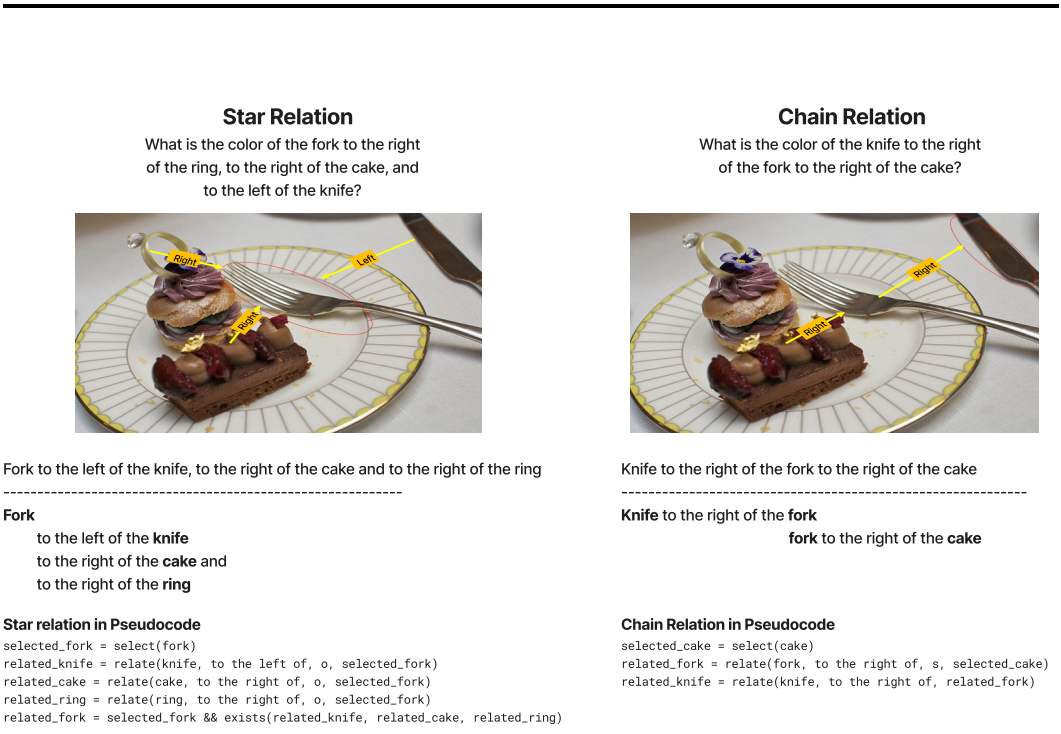


Figure 12.5: Overview of pseudocodes for two different traversal types in the VISREAS corpus.

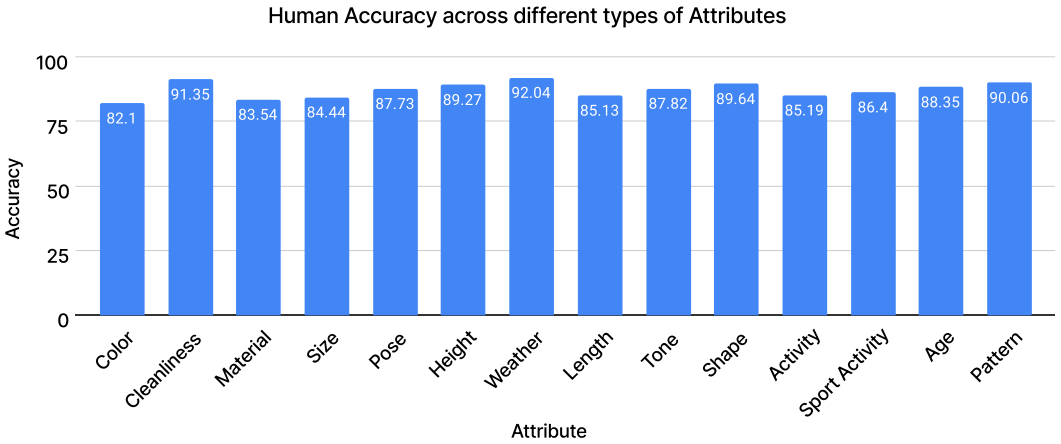


Figure 12.6: Human accuracy on different attributed questions

settings without temperature tuning, nucleus sampling, repetition penalty, etc. Specific settings used for zeroshot and finetuning are presented below:

VisProg. The original VisProg script uses `text-davinci-003` model which is around 10 times more expensive than `gpt-3.5-turbo` model. To cut evaluation costs, we use the `gpt-3.5-turbo` model instead. All 20 examples are found in GQA evaluation script for code generation.

ViperGPT. For similar reason as VisProg, we use `gpt-3.5-turbo` model for code generation to reduce costs. Since generated code doesn't always return functional Python code, we return either "None" or "ERROR" in these cases. In cases where the code throws an error, the

answer defaults to “ERROR”. In cases where the code didn’t have a return statement, the answer defaults to “None”.

LOGIC2VISION. The effective batch size is kept at 4 across experiments. LoRA modules are only attached to query and value linear layers in attention layers. The batch size and gradient accumulation steps are adjusted accordingly. Due to memory requirements, we set batch size to 1 on each GPU and set gradient accumulation steps to 4. We’ve have used 2-4 A6000 GPUs with distributed data parallel (DDP) strategy for multi-GPU training. Training LOGIC2VISION on VISREAS takes around 13 hours using 2 A6000 with LLaVA-1.5 backbone.

Hyperparameters	Values
Effective batch size	4
Learning rate	5e-6 (GQA), 2e-5 (VISREAS)
Precision	bfloat16
Optimiser	AdamW
Schedule	Linear warmup with cosine decay
Warmup steps	128
Epoch	1

Table 12.1: Hyperparameters for LOGIC2VISION model

InstructBLIP / BLIP-2 / LLaVA-1.5. On GQA, we use identical configuration as for LLaVA-1.5. For InstructBLIP and BLIP-2, we observe that batch size of 4 causes the model to output repetitive tokens during inference. For that reason, we increase the effective batch size to 8. We use the same original prompt that the authors have reported in their original papers.

On VISREAS, we again use identical configuration as for LLaVA-1.5. For InstructBLIP and BLIP-2, we lower the learning rate to 5e-6 and increase the effective batch size to 8 for the same reason above.

LXMERT / ViLBERT / CRF. For all three models trained with the classification task, we used the default hyperparameters that have been used to finetune on GQA corpus for consistency. As GQA and VISREAS share the same image and scenegraphs, using the same model with the same configuration should produce different results if the two tasks are different. And the result section reflects the distinction between GQA and VISREAS.

Hyperparameters	Values
Rank	8
Alpha	16
Dropout	0.05

Table 12.2: LoRA configurations

Hyperparameters	LXMERT	ViLBERT	CRF
Learning rate	1e-5	0.00004	1e-4
Optimizer	BertAdam	AdamW	BertAdam
Schedule	Linear Warmup	Linear Warmup	Linear Warmup
Epoch	4	20	13

Table 12.3: Hyperparameters of all CLS baselines

12.4 Effect of pseudocode finetuning

We study the effect of finetuning a VLM to perform VQA through pseudocode-guided reasoning. Table 12.4 demonstrates that finetuning LLaVA-1.5 to follow pseudocode consistently improves performance on VISREAS for both 7B and 13B models.

Model size	Without Pseudocode	With Pseudocode
7B	57.36	62.74
13B	61.38	66.20

Table 12.4: Effect of pseudocode finetuning on LLaVA-1.5

12.5 Examples from VISREAS and GQA

In Figure 12.7, we show example questions from VISREAS and GQA using the same image. In general, VISREAS tends to have longer questions compared to GQA. Additionally, VISREAS questions involve more than two objects, whereas GQA primarily centers on one or two objects.

12.6 Mechanical Turk Details

To evaluate human performance, we used Amazon Mechanical Turk to collect human responses for 5000 random questions, taking a majority vote among three workers for each question. We limited our pool of crowdworkers to individuals located in the US or Canada, requiring a minimum of 1,000 previously approved HITs with a 95% approval rate. Additionally, participants had to achieve a minimum score of 70% or higher on our qualification task before gaining access to our main task. In the subsequent sections, we provide details of this response collection process.

Qualification Test for Worker Selection To secure accurate human assessments, we carefully designed a qualification test using Amazon Mechanical Turk interfaces (Figure 12.8). This test aimed to select proficient workers capable of accurately completing the VISREAS task: (1) The qualification test encompassed two distinct tasks. The initial task focused on careful comprehension of instructions. Workers were required to attentively read the instructions and subsequently answer a set of multiple-choice questions to assess their grasp of the task’s nuances. (2) Upon successful completion of the first task, the qualified workers proceeded to the task proficiency evaluation stage. Here, a series of ten questions, each accompanied by an image, were presented. The workers’ task was to select the correct answer from a dropdown list of 2013 entries. The selection process for the final evaluation cohort prioritized workers who achieved correct answers for more than seven out of the ten questions.

Human Accuracy Assessment Interfaces After gathering qualified workers who are aware and proficient in our task, we move to the final stage of the evaluation process (Figure 12.9). For each Human Intelligence Task (HIT), an image and the corresponding question were provided. Workers were tasked with selecting the correct answer from the same dropdown list used for the worker selection stage. Furthermore, we requested workers to rate the complexity and structural integrity of the presented question, thereby acquiring insights into the inherent challenges posed by various question types.

To facilitate a deeper understanding of the potential issues with the queries, we encouraged workers to provide additional details about any perceived problems. If a worker identified a problematic aspect within the question, they were encouraged to rephrase or rewrite








	<p>VisReas</p> <p>Question: Are the doll and the soda bottle found sitting on or standing on the armchair in the image? Answer: The question itself problematic Explanation: There is no armchair present in the picture. Category: verify.rel</p>	<p>GQA</p> <p>Question: What kind of furniture is the doll to the left of the figurine sitting on? Answer: Table Category: query.obj</p>
	<p>VisReas</p> <p>Question: What is the common attribute of the pole, the road sign and the leaves which are to the right of the store in the picture? Answer: Green Category: query.attr</p>	<p>GQA</p> <p>Question: Are the cars on the left or on the right side of the photo? Answer: Right Category: choose.rel</p>
	<p>VisReas</p> <p>Question: Among the floor, the doorway to the left of the red graffiti and the door, how many things are made of concrete? Answer: Two Category: count.attr</p>	<p>GQA</p> <p>Question: What is the floor made of? Answer: Concrete Category: query.attr</p>
	<p>VisReas</p> <p>Question: Do the drawer and the floor to the right of the white shoes and to the left of the white dishwasher share the same material? Answer: The question itself is problematic Explanation: The floor is to the left of the white shoes and to the left of the white dishwasher Category: compare.attr</p>	<p>GQA</p> <p>Question: What is common to the drawer and the floor? Answer: Material Category: query.attr</p>
	<p>VisReas</p> <p>Question: Among knife, napkin, crust and wall, what object in the image has the same color as the plate and the coffee cup? Answer: Napkin Category: query.obj</p>	<p>GQA</p> <p>Question: Is the coffee cup tall and white? Answer: Yes Category: verify.attr</p>
	<p>VisReas</p> <p>Question: Do you see any tiny stop sign on the large and metal post and any large flower? Answer: No Category: verify.attr</p>	<p>GQA</p> <p>Question: What's on the post? Answer: Stop sign Category: query.obj</p>
	<p>VisReas</p> <p>Question: What are the soap bottle, the bench and the pole in front of the brown trees made of in the image? Answer: The question itself is problematic Explanation: There are no soap bottle and bench present in the photo Category: query.attr</p>	<p>GQA</p> <p>Question: Are there any fences? Answer: Yes Category: verify.obj</p>

Figure 12.7: Example questions from the VISREAS and the GQA corpuses.

the query to address the issue. This dynamic engagement aimed to uncover underlying complexities and refine the evaluation process.

12.7 SELF-IMAGINE Prompts

The prompt used for image generation is shown in Listing 12.1. Please see the code repository for the complete prompt.

```

1
2 Q: Alfie, the albatross, flies 400 kilometers every day. If the
   circumference of the earth is 40,000 kilometers, how many days will
   it take Alfie to fly a distance equal to half of the way around the
   earth?
3
4 # HTML code:
5
6
7 <!DOCTYPE html>
8 <html lang="en">
9 <head>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <title>Alfie's Journey</title>
13 <style>
14   .diagram-container {{
15     display: flex;
16     align-items: center;
17     justify-content: center;
18     flex-direction: column;
19     font-family: Arial, sans-serif;
20   }}
21
22   .earth {{
23     position: relative;
24     width: 200px;
25     height: 200px;
26     border: 3px solid blue;
27     border-radius: 50%;
28     overflow: hidden;
29   }}
30
31
32
33   .text {{
34     margin: 10px;
35     text-align: center;
36   }}
37
38   .stat {{
39     display: flex;
40     justify-content: space-around;
41     margin-top: 20px;
42   }}
43
44   .stat > div {{
45     text-align: center;
46   }}
47 </style>
48 </head>
49 <body>
50
51 <div class="diagram-container">
52   <div class="earth">
53     <div class="albatross-flight"></div>
54   </div>
55   <div class="text">Alfie's Journey Around the Earth</div>
56   <div class="stat">
57     <div>
58       <strong>Alfie's Daily Distance:</strong><br>
59       400 km
60     </div>
61     <div>
62       <strong>Earth's Circumference:</strong><br>

```

```
63     40,000 km
64   </div>
65   <div>
66     <strong>Target Distance:</strong><br>
67     20,000 km (halfway around the Earth)
68   </div>
69 </div>
70 </div>
71
72 </body>
73 </html>
```

Listing 12.1: Prompt for generating HTML using VLM

12.8 Prompts and Datasets for MIND

12.8.1 Prompts for Conversation

TWO PROFESSORS

Convert the context above as a multi-turn discussions between two professors. Make sure that their discussions strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

TEACHER STUDENT

Convert the context above as a multi-turn discussions between a teacher and a student. The student has questions about the context and the teacher solves each of them step-by-step. Make sure that their discussions strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

TWO STUDENTS

Convert the context above as a multi-turn discussions between two students who are working on their assignment related to the given context. Make sure that their discussions strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

INTERVIEW

Conduct an interview-style conversation where one participant acts as the interviewer, asking questions exclusively related to the content provided, while the other participant serves as the subject matter expert, providing detailed responses based on the content. Make sure that their discussions strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

Task	Prompt	
	Question Only	Question + Image
GSM8K, ASDIV, SVAMP, MAWPS	Solve the math problem. Think step-by-step. Always end your answer with ‘The answer is $\langle answer \rangle$ ’.	Solve the math problem using the image. Think step-by-step. Always end your answer with ‘The answer is $\langle answer \rangle$ ’.
PENGUINS IN A TABLE	Answer questions about a table of penguins and their attributes.	Answer questions about a table of penguins and their attributes using the image.
COLORED OBJECTS	Answer extremely simple questions about the colors of objects on a surface.	Answer extremely simple questions about the colors of objects on a surface using the image.
OBJECT COUNTING	Questions that involve enumerating objects and asking the model to count them.	Questions that involve enumerating objects and asking the model to count them using the image.
NAVIGATE	Given a series of navigation instructions, determine whether one would end up back at the starting point.	Given a series of navigation instructions, determine whether one would end up back at the starting point using the image.
DATE UNDERSTANDING	Infer the date from context.	Infer the date from context using the image.
GEOMETRIC SHAPES	Name geometric shapes from their SVG paths.	Name geometric shapes from their SVG paths and using the image.
TEACHER STUDENT	Answer questions about which times certain events could have occurred.	Answer questions about which times certain events could have occurred using the image.
TRACKING SHUFFLED OBJECTS	A task requiring determining the final positions of a set of objects given their initial positions and a description of a sequence of swaps.	A task requiring determining the final positions of a set of objects given their initial positions and a description of a sequence of swaps using the image.

Table 12.5: Prompts used for both reasoning and mathematics tasks. For all reasoning tasks, we also add *Please think step-by-step, and finally answer by selecting an option using the format “The answer is $\langle option \rangle$ ”* after adding the question to the above mentioned prompts.

PROBLEM SOLVING

Question	John buys twice as many red ties as blue ties. The red ties cost 50% more than blue ties. He spent \$200 on blue ties that cost \$40 each. How much did he spend on ties?
Image	<p>John's Tie Purchase</p> <p>Variables and Values</p> <ul style="list-style-type: none"> • Number of blue ties: b • Number of red ties: $2b$ • Cost of a blue tie: \$40 • Cost of a red tie: $\\$40 * 1.5 = \\60 • Total cost of blue ties: $b * \\$40$ • Total cost of red ties: $2b * \\$60$ • Total cost of ties: $(b * \\$40) + (2b * \\$60)$ <p>Calculations</p> <p>Given that John spent \$200 on blue ties:</p> $b * \$40 = \200 <p>Solving for b:</p> $b = \$200 / \40 $b = 5$ <p>Therefore, John bought 5 blue ties and $2 * 5 = 10$ red ties.</p> <p>The total cost of blue ties is $5 * \\$40 = \\200.</p> <p>The total cost of red ties is $10 * \\$60 = \\600.</p> <p>Total cost of ties: $\\$200 + \\$600 = \\$800$</p>
Label	800
Output With Image	<p>Let the number of blue ties be b. The number of red ties is twice the number of blue ties, so there are $2b$ red ties. The cost of a blue tie is \$40, and the cost of a red tie is 50% more than the cost of a blue tie, so the cost of a red tie is $40 * 1.5 = \\$60$. The total cost of the blue ties is $b * 40$, and the total cost of the red ties is $2b * 60$. The total cost of the ties is $b * 40 + 2b * 60$. We know that John spent \$200 on blue ties, so we can write the equation $b * 40 = 200$. Solving for b, we get $b = 5$. Therefore, John bought 5 blue ties and $2 * 5 = 10$ red ties. The total cost of the blue ties is $5 * 40 = \\$200$. The total cost of the red ties is $10 * 60 = \\$600$. The total cost of the ties is $200 + 600 = \\$800$. The answer is \$800.</p>
Output Without Image	<p>John bought 5 blue ties and 10 red ties. The blue ties cost \$200, so the red ties cost \$300. He spent a total of \$500 on ties. The answer is 500.</p>

Table 12.6: Example of Image improving reasoning in GSM8K task for GEMINI PRO.

Convert the context above as a multi-turn problem-solving conversation where participants analyze challenges or scenarios presented in the content and brainstorm solutions within the context of the provided material, avoiding speculation or unrelated discussions. Make sure that their conversation strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

LAYMAN KNOW-ALL

Imagine you are presenting the content above step-by-step to a layman. While you are presenting, the layman has a lot of followup questions regarding your presentation. You answer the questions step-by-step with chain-of-thoughts. Design this interaction between you and the layman as a multi-turn conversational manner. Make sure that the interaction strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

DEBATE

Question	Our class got fifty-four books from the library. Then we got twenty-three more books from the library. How many books did our class get from the library?
Image	
Label	77
Output With Image	To solve this problem, we need to add the initial number of books (50) and the number of books received from the library (23). $50 + 23 = 73$ The answer is 73.
Output Without Image	To solve this problem, we need to add the number of books the class got from the library initially (54) to the number of books they got from the library later (23). Step 1: Add 54 and 23 to find the total number of books the class got from the library. $54 + 23 = 77$ The answer is 77.

Table 12.7: Example of Image hurting reasoning in ASDIV task for LLAVA-1.5.

Convert the context above as a multi-turn debate-style conversation where the participants present arguments and counterarguments based solely on the content provided, without introducing external information or personal opinions. Each participant defends others arguments step-by-step with chain-of-thoughts. Make sure that the conversation strictly adhere to the context above and remains faithful to information in the context. Please DONOT add any new information/reference other than the context.

12.8.2 Evaluation Metric Details

We evaluate the LLM trained on raw and synthetic data using ten diverse general reasoning tasks, three mathematical tasks and one specialized knowledge tasks.

General Purpose Reasoning Tasks. All the benchmarks under this category are evaluated in zero-shot manner.

- **ARC Easy (ARC-E) and ARC Challenge (ARC-C)** (Clark et al., 2018): This dataset is proposed by the AI2 Reasoning Challenge (ARC). There are two sets of this data: (1) ARC-E and (2) ARC-C, containing science exam questions from grades 3 to 9. The ARC Challenge set includes more difficult questions compared to ARC-E that necessitate higher-order reasoning.

Question	<p>Alice, Bob, and Claire are on the same team in a soccer match. At the start of the match, they are each assigned to a position: Alice is playing goalkeeper, Bob is playing left winger, and Claire is playing center midfielder.</p> <p>As the game progresses, pairs of players occasionally swap positions. First, Claire and Bob trade positions. Then, Alice and Bob trade positions. Finally, Claire and Alice trade positions. At the end of the match, Alice is playing</p> <p>Options: (A) goalkeeper (B) left winger (C) center midfielder</p>
Image	<p>Soccer Positions</p> <p>At the start of the match, the positions are:</p> <ul style="list-style-type: none"> • Alice: goalkeeper • Bob: left winger • Claire: center midfielder <p>After the first trade, the positions are:</p> <ul style="list-style-type: none"> • Alice: goalkeeper • Bob: center midfielder • Claire: left winger <p>After the second trade, the positions are:</p> <ul style="list-style-type: none"> • Alice: left winger • Bob: goalkeeper • Claire: center midfielder <p>After the third trade, the positions are:</p> <ul style="list-style-type: none"> • Alice: center midfielder • Bob: goalkeeper • Claire: left winger <p>At the end of the match, Alice is playing center midfielder.</p>
Label	(B)
Output With Image	<p>After the first trade, Alice is still the goalkeeper, Bob is now the center midfielder, and Claire is the left winger.</p> <p>After the second trade, Alice is now the left winger, Bob is the goalkeeper, and Claire is still the center midfielder. After the third trade, Alice is the center midfielder, Bob is the left winger, and Claire is the goalkeeper. At the end of the match, Alice is playing center midfielder. The answer is (C).</p>
Output Without Image	<p>First, Claire and Bob trade positions. This means that Claire is now playing left winger and Bob is playing center midfielder. Next, Alice and Bob trade positions. This means that Alice is now playing center midfielder and Bob is playing goalkeeper. Finally, Claire and Alice trade positions. This means that Claire is now playing goalkeeper and Alice is playing left winger. Therefore, at the end of the match, Alice is playing left winger. The answer is (B).</p>

Table 12.8: Example of Image hurting reasoning in TRACKING SHUFFLED OBJECTS of three objects task for GEMINI PRO.

- **RACE** (Lai et al., 2017): This dataset has been collected from English reading comprehension exams designed for middle and high school Chinese students.
- **PIQA** (Bisk et al., 2020): Physical Interaction Question Answering evaluates physical commonsense reasoning ability of the language model.
- **Winogrande** [Wino.](Sakaguchi et al., 2019): This benchmark is structured as a fill-in-the-blank task with binary options, requiring the LLM to select the correct option for a given sentence, primarily focusing on commonsense reasoning and pronoun disambiguation tasks.

Question	If you follow these instructions, do you return to the starting point? Take 8 steps. Turn around. Take 8 steps. Options: - Yes - No
Image	Return to Starting Point Instructions: Take 8 steps. Turn around. Take 8 steps. Answer: No Explanation: After following the instructions, you will be facing the opposite direction from where you started. You would need to turn around again to face the starting point.
Label	(B)
Output With Image	The answer is No . You start by taking 8 steps forward. Then you turn around, which means you are now facing the opposite direction. Then you take 8 steps forward again. This means you are now 8 steps away from your starting point, but facing the opposite direction. You would need to turn around again to face the starting point.
Output Without Image	The answer is yes .

Table 12.9: Example of Image hurting reasoning in NAVIGATE task for GEMINI PRO.

- **HellaSwag** (Zellers et al., 2019): This dataset evaluates a model’s ability to resolve scenarios in a way that is both contextually appropriate and logically consistent, testing its grasp of language comprehension and commonsense reasoning.
- **OpenBookQA** [OBQA] (Mihaylov et al., 2018): This dataset is designed to evaluate deeper understanding of elementary science facts by requiring models to apply these facts to novel situations using both open book knowledge and external commonsense reasoning.
- **TruthfulQA** [TFQA] (Lin et al., 2022): Evaluates models’ ability to generate factually correct answers by presenting 817 questions across 38 categories, designed to challenge common misconceptions.
- **CommonSenseQA** [CSQA] (Talmor et al., 2019): This dataset has been designed to test commonsense reasoning through multiple-choice questions created from CONCEPT-NET (Speer et al., 2017) relations, which requires prior knowledge beyond contextual associations for accurate answering.
- **Social-IQA** [SIQA] (Sap et al., 2019): Evaluates LLM’s ability to reason about people’s actions and their social implications.

Math and Specialized Knowledge Tasks. For these tasks, we evaluate the LLM in few-shot manner.

- **GSM8K** (Cobbe et al., 2021a): This benchmark comprises of high quality linguistically diverse grade school math word problems that evaluates the multi-step and logical reasoning ability of LLM. In this setup, we prompt the LLM with eight chain-of-thought examples from Wei et al. (2022) and take the majority vote of the answers from greedy decoding following the approach in Wang et al. (2022).

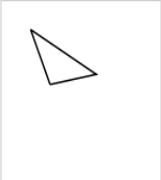
Question	<p>This SVG path element <code><path d="M 30.17,45.97 L 58.79,40.36 L 18.10,15.70 M 18.10,15.70 L 30.17,45.97"/></code> draws a</p> <p>Options:</p> <p>(A) circle (B) heptagon (C) hexagon (D) kite (E) line (F) octagon (G) pentagon (H) rectangle (I) sector (J) triangle</p>
Image	<p>What does this SVG path draw?</p> 
Label	(J)
Output With Image	<p>The path starts at point (30.17,45.97) and draws a line to point (58.79,40.36). Then it draws another line to point (18.10,15.70). Finally, it draws a line back to the starting point (30.17,45.97). The resulting shape is a triangle. The answer is (J).</p>
Output Without Image	<p>The path element consists of two subpaths. The first subpath starts at (30.17, 45.97) and goes to (58.79, 40.36), then (18.10, 15.70). The second subpath starts at (18.10, 15.70) and goes back to (30.17, 45.97). The first subpath looks like a line from (30.17, 45.97) to (18.10, 15.70), and the second subpath looks like a line from (18.10, 15.70) to (30.17, 45.97). So it should be a line. The answer is (E).</p>

Table 12.10: Example of Image improving reasoning in GEOMETRIC SHAPES task for GEMINI PRO.

- **MATH** (Hendrycks et al., 2021e): This dataset contains challenging competition mathematics problems that requires step-by-step processing of the problem to derive the solution. We choose 4-shot prompt from Lewkowycz et al. (2022b) for our evaluation process.
- **MMLU** (Hendrycks et al., 2021b): This task is designed to evaluate a LLM’s multitask accuracy across 57 diverse subjects, including elementary mathematics, US history, and law in multiple-choice question format, requiring extensive world knowledge and problem-solving skills for high performance. We explicitly consider MMLU-STEM as it contains comprehensive math and science problems that requires multi-hop and complex reasoning ability. Using the evaluation pipeline of LM Eval Harness, we evaluate the LLM with 5-shot prompts for this task.

Question	A pound of almonds costs \$10 while a pound of walnuts costs \$15. How much more does it cost for a mixture of 1/2 pound almonds and 1/3 pound walnuts than a mixture of 1/5 pound almonds and 1/3 pound walnuts?
Image	<p>The diagram shows a sequence of boxes connected by mathematical symbols. The first box is labeled 'Almonds' with '1/2 lb' and '\$10'. This is followed by a '+' sign, a second box labeled 'Walnuts' with '1/3 lb' and '\$15'. This is followed by an '=' sign, a third box labeled 'Mixture' with '?' and '?'. This is followed by a '-' sign, a fourth box labeled 'Almonds' with '1/5 lb' and '\$10'. This is followed by an '=' sign, and a final box labeled 'Difference' with '?'.</p>
Label	3
Output With Image	<p>To solve this problem, we need to find the cost of each mixture and then compare them.</p> <p>Cost of 1/2 pound almonds and 1/3 pound walnuts: We know that a pound of almonds costs \$10, and a pound of walnuts costs \$15. To find the cost of 1/2 pound almonds and 1/3 pound walnuts, we need to multiply the cost per pound by the amount of each nut in the mixture: $(10/\text{lb}) * (1/2 \text{ lb almonds}) + (15/\text{lb}) * (1/3 \text{ lb walnuts}) = \\$10 + \\$5 = \\15</p> <p>Cost of 1/5 pound almonds and 1/3 pound walnuts: We can find the cost of this mixture in a similar way: $(10/\text{lb}) * (1/5 \text{ lb almonds}) + (15/\text{lb}) * (1/3 \text{ lb walnuts}) = \\$2 + \\$5 = \\7</p> <p>Comparison: Now we can compare the costs of the two mixtures: $\\$15 - \\$7 = \\$8$</p> <p>The answer is \$8.</p>
Output Without Image	<p>To solve this problem, we need to find the cost of the mixture of 1/2 pound almonds and 1/3 pound walnuts and compare it to the cost of the mixture of 1/5 pound almonds and 1/3 pound walnuts.</p> <p>First, we need to find the cost of 1/2 pound of almonds and 1/3 pound of walnuts. We can do this by multiplying the cost per pound of almonds and walnuts by the respective weights:</p> <p>1/2 pound of almonds * \$10/pound = $\\$1/2 * 10 = 5$ 1/3 pound of walnuts * \$15/pound = $\\$1/3 * 15 = 5$</p> <p>The total cost of the mixture is the sum of the costs of the almonds and walnuts:</p> <p>$\\$5 + \\$5 = \\$5 + 5 = 10$</p> <p>Next, we need to find the cost of 1/5 pound of almonds and 1/3 pound of walnuts. We can do this in the same way:</p> <p>1/5 pound of almonds * \$10/pound = $\\$1/5 * 10 = 2$ 1/3 pound of walnuts * \$15/pound = $\\$1/3 * 15 = 5$</p> <p>The total cost of the mixture is the sum of the costs of the almonds and walnuts:</p> <p>$\\$2 + \\$5 = \\$2 + 5 = 7$</p> <p>Now we can compare the costs of the two mixtures:</p> <p>$\\$10 - \\$7 = \\$10 - 7 = 3$</p> <p>The answer is \$3.</p>

Table 12.11: Example of Image hurting reasoning in GSM8K task for LLaVA-1.5.

12.9 Additional Experiments and Results

12.9.1 Results of Pretraining LLM from Scratch

We pretrain a 8B LLM from scratch with 300B tokens using (i) 4 snapshots of CommonCrawl (ii) OWM-4B and (iii) wikipedia, books and epubs corpus corresponding to 486B, 4B and 84B original tokens respectively. To emphasize math over other datasets, we provide 8 epochs of OWM-4B in the pretraining blend resulting in 35B OWM tokens that will be seen by the LLM during pretraining. For all other datasets, we maintain 0.46 epochs. For our experimentation with synthetic corpus, we analyze four variations in the OWM corpus while keeping the other data constant:

Question	If you follow these instructions, do you return to the starting point? Always face forward. Take 5 steps forward. Take 8 steps backward. Take 4 steps forward. Take 4 steps right. Options: - Yes - No
Image	No, you will not return to the starting point if you follow these instructions. The sequence of steps is as follows: 1. 7 steps left: 7 steps to the left 2. 2 steps backward: 2 steps backward 3. 7 steps backward: 7 steps backward 4. 7 steps backward: 7 steps backward 5. 3 steps forward: 3 steps forward After completing these steps, you will be 7 steps to the left of your starting position, and then you will take 2 steps backward, which will put you 2 steps behind your starting position. You will then take 7 steps backward, which will put you 7 steps further behind your starting position. Finally, you will take 3 steps forward, which will bring you 3 steps closer to your starting position. However, you will still be 4 steps behind your starting position (7 steps left + 2 steps backward + 7 steps backward + 7 steps backward - 3 steps forward = 4 steps behind).
Label	No
Output With Image	The answer is No
Output Without Image	To determine whether one would end up back at the starting point, we need to analyze the given navigation instructions step-by-step. Take 5 steps forward. Take 8 steps backward. Take 4 steps forward. Take 4 steps right. Let's analyze each step: Take 5 steps forward: After taking 5 steps forward, the person would be 5 steps away from the starting point. Take 8 steps backward: After taking 8 steps backward, the person would be 8 steps away from the starting point, in the opposite direction. Take 4 steps forward: After taking 4 steps forward, the person would be 4 steps away from the starting point, but now they are moving in the correct direction. Take 4 steps right: After taking 4 steps right, the person would be 4 steps away from the starting point, but now they are at a right angle to the starting point. Since the person is now at a right angle to the starting point and moving in the correct direction, they will eventually return to the starting point by continuing to move forward. The answer is: Yes

Table 12.12: Example of Image improving reasoning in NAVIGATE task for LLAVA-1.5.

- **MIND-OWM-4B [TWO STUDENTS]**. This data includes conversations between two students.
- **OWM-4B + MIND-OWM-4B [1:1]**. We sample raw and synthetic conversations in a 1:1 ratio, ensuring an equal number of tokens to be seen during pretraining from both sources. For the synthetic data, we utilize the TWO STUDENTS conversations.
- **OWM-4B + MIND-OWM-4B [Concat]**. We concatenate each raw context with all seven synthetic conversations sequentially.
- **MIND-OWM-4B [Longest Conversation]**. From the seven conversations generated for each context, we select the longest conversation in token count.

As shown in [Table 12.14](#), conversational synthetic data improves general purpose reasoning ability of LLM. Specifically, the concatenation of raw text and conversations yields the best average score for all combinations—highlighting the efficacy of both data towards generalizability of LLM across wide range of reasoning tasks.

In addition, for mathematical benchmarks, only synthetic data produce the best improvement over the raw data ([Table 12.15](#)). The nature of conversational data being composite and structured helps the LLM to perform well in tasks that requires step-by-step processing of a complex problem. Conversely, specialized knowledge tasks require both raw and synthetic data to attain the maximum gain.

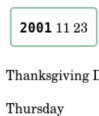
Question	<p>In the US, Thanksgiving is on the fourth Thursday of November. Today is the US Thanksgiving of 2001. What is the date yesterday in MM/DD/YYYY?</p> <p>Options:</p> <p>(A) 09/12/2002 (B) 11/30/2002 (C) 11/21/2002 (D) 11/21/2076 (E) 11/07/2002 (F) 11/15/2002</p>
Image	 <p>2001 11 23</p> <p>Thanksgiving Day</p> <p>Thursday</p>
Label	(C)
Output With Image	The answer is (E) 11/07/2002.
Output Without Image	<p>To infer the date from context, we need to determine the date of US Thanksgiving in 2001.</p> <p>Step 1: Determine the date of US Thanksgiving in 2001. The US Thanksgiving holiday is celebrated on the fourth Thursday of November. In 2001, November has 30 days.</p> <p>Step 2: Calculate the date of US Thanksgiving in 2001. 4th Thursday of November 2001: Thursday = 22 (since it's a Thursday) November = 30 22 + 30 = 52 (the date in numerical format)</p> <p>Step 3: Convert the numerical date to MM/DD/YYYY format. 52 = 11/22/2001</p> <p>The answer is (C) 11/21/2002.</p>

Table 12.13: Example of Image hurting reasoning in DATE UNDERSTANDING task for LLAVA-1.5.

Dataset	ARC-E	Race	PIQA	Wino.	HellaSwag	ARC-C	OBQA	TFQA	CSQA	SIQA	Avg-All
OWM-4B	66.79	35.98	77.69	62.19	68.23	38.91	37.20	35.92	19.57	44.42	48.69
MIND-OWM-4B [TWO STUDENTS]	68.14	36.75	77.86	63.06	69.11	40.19	39.40	37.80	19.66	45.55	49.75
OWM-4B+MIND-OWM-4B [1:1]	69.74	37.32	77.64	63.69	69.51	40.87	38.20	34.97	20.39	44.47	49.68
OWM-4B+MIND-OWM-4B [Concat]	69.28	38.37	78.02	64.09	68.66	39.76	39.00	38.38	22.52	44.63	50.27
MIND-OWM-4B [Longest Conversation]	68.39	36.75	77.64	62.04	68.91	40.02	39.40	38.23	20.23	44.52	49.61

Table 12.14: Evaluation of 8B LLM on General Reasoning tasks: Conversations provide improvement over raw data in general purpose reasoning tasks including commonsense, factual and social reasoning tasks.

Dataset	GSM8K	MATH	MMLU-STEM	MMLU-Humanities	MMLU-Social-Sciences	MMLU-Others	MMLU	Avg-All
OWM-4B	4.78	4.92	26.29	25.93	26.75	27.16	26.46	12.05
MIND-OWM-4B [TWO STUDENTS]	10.77	5.30	26.93	26.78	26.81	27.87	27.06	14.38
OWM-4B+MIND-OWM-4B [1:1]	8.49	5.02	28.01	28.44	28.40	28.39	28.32	13.94
OWM-4B+MIND-OWM-4B [Concat]	8.04	4.98	29.18	29.22	29.51	31.54	29.79	14.27
MIND-OWM-4B [Longest Conversation]	8.57	4.60	26.77	27.16	29.12	29.29	27.97	13.71

Table 12.15: Evaluation of 8B LLM on Math and Specialized Knowledge tasks: Conversations are always beneficial for mathematical reasoning where specialized knowledge tasks further benefit from having both raw and synthetic data in the corpus.

12.9.2 Breakdown of Individual Tasks Results of Continued Pretrained LLM

In this section, we further breakdown the performance of models trained on individual and combinations of conversation styles across general purpose reasoning tasks and specialized knowledge tasks.

Performance across Individual Prompt Style. As shown in Table 12.16, synthetic data overall achieves highest gain for general purpose reasoning task compared to using raw or rephrased data. Table 12.17 further validates the efficacy of synthetic conversations on mathematical reasoning tasks where model trained with all styles of conversational data generated from OWM-4B gets the highest gain across all other models—highlighting the potential of upsampling high-quality data by generating synthetic data of diverse styles using a small seed data.

Dataset	Style	ARC-E	Race	PIQA	Wino.	HellaSwag	ARC-C	OBQA	TFQA	CSQA	SIQA	Avg-All
OWM-4B	Raw	71.89	37.89	78.24	65.98	71.42	46.33	41.40	36.96	32.35	46.57	52.90
Rephrase-OWM-4B	Rephrase	72.05	38.28	78.07	63.14	71.16	45.31	42.20	47.09	33.33	45.19	53.58
MIND-OWM-4B	TWO PROFESSORS	72.18	38.85	77.20	66.38	71.54	44.20	40.40	42.51	32.35	46.47	53.21
	TEACHER STUDENT	75.17	38.76	78.35	66.46	72.08	47.70	40.20	44.88	38.74	46.06	54.84
	TWO STUDENTS	72.90	38.56	78.24	65.82	72.24	46.67	41.00	44.10	38.25	45.45	54.32
	LAYMAN KNOWALL	74.12	39.04	78.45	65.27	72.19	46.42	41.00	46.25	41.28	44.88	54.89
	DEBATE	74.92	38.37	78.45	65.75	71.89	47.78	40.40	45.47	38.41	46.16	54.76
	INTERVIEW	73.82	37.99	78.13	65.11	72.18	48.72	42.00	47.81	36.04	45.45	54.73
	PROBLEM SOLVING	74.41	38.37	78.07	65.59	71.67	49.40	41.20	47.04	37.02	46.26	54.90
	LONGEST CONVERSATION	74.71	37.99	78.18	64.80	72.10	47.61	41.40	45.49	39.80	46.52	54.86
ALL CONVERSATIONS	75.17	39.04	77.86	65.43	72.31	49.40	41.00	46.68	40.79	46.42	55.41	
OWM-4B+MIND-OWM-4B [1:1]	Combination	74.12	37.99	78.18	66.54	72.28	48.12	41.40	39.27	40.70	46.37	54.50
OWM-4B+MIND-OWM-4B [Concat]		74.92	38.28	77.58	67.32	72.63	48.55	41.80	42.26	40.95	46.72	55.10

Table 12.16: **Results of 7B LLM on General Reasoning Tasks:** We evaluate both the baseline and model trained with synthetic data across diverse tasks that focus on general reasoning, language understanding and commonsense.

Dataset	Style	GSM8K	MATH	MMLU-STEM	MMLU-Humanities	MMLU-Social-Sciences	MMLU-Others	MMLU	Avg-All
OWM-4B	Raw	12.96	4.92	39.39	41.15	52.84	52.85	45.91	21.26
Rephrase-OWM-4B	Rephrase	11.68	5.46	39.71	40.77	54.76	52.40	46.17	21.10
MIND-OWM-4B	TWO PROFESSORS	13.50	4.52	37.93	41.89	52.32	50.76	45.25	21.09
	TEACHER STUDENT	22.74	5.96	40.72	42.21	56.78	55.13	47.93	25.54
	TWO STUDENTS	21.30	6.20	41.90	43.40	57.07	55.65	48.77	25.42
	LAYMAN KNOWALL	17.74	5.46	41.96	44.27	56.19	55.62	48.87	24.02
	DEBATE	23.96	6.12	40.18	42.40	55.38	55.33	47.61	25.90
	INTERVIEW	20.92	5.86	40.53	41.21	55.48	53.91	46.99	24.59
	PROBLEM SOLVING	24.72	6.16	41.36	42.21	55.18	55.23	47.74	26.21
	LONGEST CONVERSATION	25.78	6.30	42.72	43.53	57.52	56.90	49.37	27.15
ALL CONVERSATIONS	26.38	7.22	42.53	44.38	58.63	58.51	50.21	27.94	
OWM-4B+MIND-OWM-4B [1:1]	Combination	21.68	6.14	42.56	43.85	57.59	57.42	49.57	25.80
OWM-4B+MIND-OWM-4B [Concat]		24.49	6.22	43.67	44.87	59.21	57.16	50.46	27.06

Table 12.17: **Results of 7B LLM on Specialized Knowledge Tasks:** In this setup, we assess the domain specific knowledge of LLM specifically on mathematics, science and general knowledge. We emphasize on the GSM8K, MATH and MMLU-STEM task, as these tasks predominantly checks the mathematical reasoning ability of the LLM.

Analysis with Complete OpenWebMath. Our experiment with complete OWM-14B shows the similar trend as before. The comprehensive nature of this larger dataset continues to reinforce the advantages of synthetic data, as models trained on it also exhibit enhanced performance across both general purpose reasoning (Table 12.18) and mathematical reasoning

tasks (Table 12.18). This consistency across different dataset sizes highlights the robustness of the benefits gained from incorporating diverse conversational styles, further supporting the notion that expanding training data through synthetic means can lead to significant advancements in the capabilities of language models.

Dataset	ARC-E	Race	PIQA	Wino.	HellaSwag	ARC-C	OBQA	TFQA	CSQA	SIQA	Avg-All
Pretraining Data	70.88	38.76	78.78	67.80	73.90	43.86	42.60	41.35	29.65	44.63	53.22
OWM-14B	73.40	37.32	77.91	65.90	72.15	47.10	41.40	38.39	39.64	46.26	53.95
MIND-OWM-14B	75.84	39.52	78.56	65.67	72.38	48.55	42.80	45.06	39.89	47.08	55.54

Table 12.18: **Evaluations on General Reasoning Tasks with complete OWM-14B:** Conversational data is beneficial for general purpose reasoning tasks.

Dataset	GSM8K	MATH	MMLU-STEM	MMLU-Humanities	MMLU-Social-Sciences	MMLU-Others	MMLU	Avg-All
Pretraining Data	9.33	4.74	37.93	41.23	51.80	53.07	45.43	34.79
OWM-14B	20.47	7.24	42.82	44.48	56.61	56.78	49.49	39.70
MIND-OWM-14B	27.29	8.24	43.55	43.95	57.95	57.45	49.91	41.19

Table 12.19: **Evaluations on Math and Specialized Knowledge Tasks with complete OWM-14B:** Conversations improve mathematical reasoning over raw data.

12.10 Additional Ablations

12.10.1 Context Length vs Conversation Quality

To generate conversations, we utilize \mathcal{M} , which supports input sequences of up to 8K tokens. However, the OpenWebMath corpus, composed of mathematical web pages from Common Crawl, often contains documents exceeding this 8K token limit, leading to errors when processing them with the LLM. A straightforward approach is to split these inputs into 8K-token windows, but initial experiments with this method reveal significant drawbacks. Conversations generated from the 8K-token inputs tend to summarize the lengthy context, resulting in a loss of substantial information from the original text.

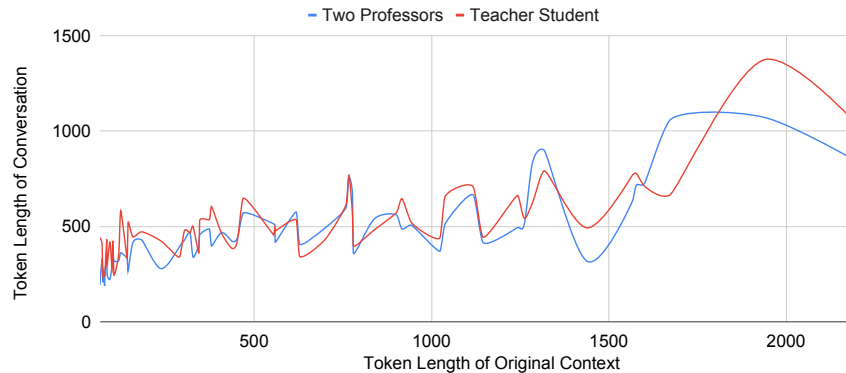


Figure 12.14: With increasing context length the generated conversation length decreases!

Therefore, we conduct an experiment on 140k samples from the OpenWebMath corpus of varying input length to determine the optimal input token length that generates conversations of following characteristics: (1) retains all relevant information from the original context, (2) remains grounded to the source material and (3) enhances the conversation with complementary explanations and reasoning. For each sample, we generate conversations using two prompt (TWO PROFESSORS and TEACHER STUDENT) and observe the token length of the generations. As depicted in Figure 12.14, with increasing input token length (X-axis), the token length of the generated conversation (Y-axis) does not scale up linearly. For example, an input context of 2K tokens results in a conversation that has 1K tokens resulting in a lot of information loss during conversion. Analyzing the Figure 12.14, we see that the input token length of 500 can generate conversation that goes beyond 500 tokens meaning that the conversation not only retains information but also adds necessary reasoning resulting in more tokens.

12.10.2 Conversation Length vs Model Performance

As shown in Table 5.1, LONGEST CONVERSATION achieves the best results among all styles. Since LONGEST CONVERSATION is curated by selecting the longest dialogue (in terms of token count) from seven conversations for a single context, it raises the question of how dialogue length impacts downstream task accuracy.

To explore the relationship between dialogue length and accuracy, we measured the average token length of dialogues across all conversational styles, including LONGEST CONVERSATION. As seen in Table 12.20, reasoning accuracy does not exhibit a linear correlation with dialogue length. For example, with PROBLEM SOLVING style we can achieve comparable accuracy to LONGEST CONVERSATION even when the average token length for PROBLEM SOLVING is 188 lower than LONGEST CONVERSATION. This highlights that the conversation length is not the only important factor to attain the maximum gain in reasoning ability. As mentioned in Section 10.5, the structure and dynamics of the conversations also play a crucial role in maximizing reasoning gains.

Style	Avg Token Length	Accuracy (Avg-All)
TWO PROFESSORS	451.95	29.12
TWO STUDENTS	452.17	32.65
PROBLEM SOLVING	465.29	33.38
TEACHER STUDENT	494.03	32.87
INTERVIEW	497.21	32.12
DEBATE	511.90	33.11
LAYMAN KNOWALL	630.23	31.74
LONGEST CONVERSATION	653.48	34.08

Table 12.20: **Conversation Length vs Downstream Task Accuracy:** Conversation length is not correlated with downstream task accuracy.

12.10.3 Conversation Quality Assessment

While the conversations generated by the LLM typically appear coherent, there are instances where the conversation fails to preserve the context or lacks grounding to the source material. In some cases, conversations may even be incomplete. Detecting poor-quality generation becomes challenging at scale. To address this, we explore two quality-filtering approaches:

Heuristic Filtering. We employ a simple heuristic based on token length. Given that the input context is limited to a maximum of 500 tokens and split into subcontexts of 500 tokens each to maximize information retention, we discard any generated conversations that fall below 50 tokens. This ensures that minimal information loss is detected early.

LLM-based Scoring. For a more comprehensive assessment, we use an LLM to score the quality of the generated conversations. We introduce four key metrics for evaluation:

- **Correctness:** Verifies that all information, such as numbers and parameters, is accurately reflected in the conversation.
- **Faithfulness:** Ensures the conversation remains grounded in the context provided.
- **Information Preservation:** Checks whether all relevant facts and knowledge from the original context are retained in the conversation.
- **New Knowledge:** Evaluates whether the conversation introduces additional explanations, reasoning, or definitions not present in the raw input.

Given a raw context and its corresponding conversation, we ask \mathcal{M} to rate the conversation on a scale of 1 to 5 in each of four metrics, with 1 representing poor quality and 5 representing the best possible conversation. To determine the overall quality, we compute the average score across the metrics and choose conversations with average scores more than or equal to 3. Additionally, we utilize the prompt from the FineWebEdu (Penedo et al., 2024) annotation framework to further check the correlation between two scoring approaches. In Figure 12.15, we plot

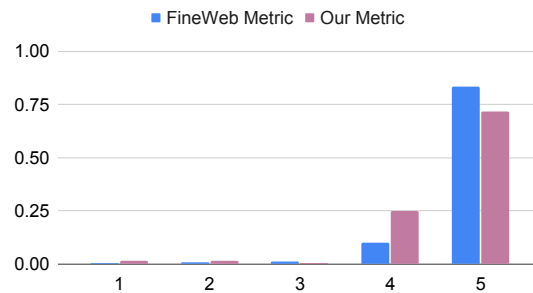


Figure 12.15: LLM tends to rate its generation higher most of the times.

the scores for 140K conversations using FineWebEdu metrics and our metrics. It is clearly visible from the figure is that LLM tends to rate its own generation higher almost all the time resulting in a skewed distribution of rating. Around 96% of conversations are labelled as high quality. However, compared to FineWebEdu, our metric results in less skewed distribution—making our approach more suitable for evaluating synthetic data derived from a seed corpus.

To further investigate, we choose 20 contexts and their corresponding conversations and manually label them on the above four metrics. We later pass these samples to LLM to obtain the quality scores. The correctness and faithfulness metrics were consistently high, with LLM showing a generation correct 96% of times and human annotators labeling a conversation correct 98% of times (with spearman correlation between two being 0.82) which validates the quality and reliability of the generated synthetic dialogues. When comparing the overall human scores with those from the LLM across the four metrics, we observe a weak correlation between two sets (Spearman’s $\rho = 0.03$) and the reasoning behind them. Human annotators prioritized the information preservation metric, while the LLM often overlooked minor information loss. Additionally, the interpretation of “New Knowledge” differed between humans and the LLM. Humans valued extra reasoning and explanation as forms of new knowledge, whereas the LLM assigned higher “New Knowledge” scores to conversations containing out-of-context information that is difficult to verify. Given these differences in the results from human and LLM-based quality filtering, we use simple heuristic filtering in this study and plan to explore other approaches in the future.

12.10.4 Compare with DEEPSEEK MATH

To assess the quality of our data, we run pre-training experiments to compare MIND-OWM with the recently released DEEPSEEK MATH (Zhihong Shao, 2024). The DEEPSEEK MATH approach is iterative. They construct a dataset for binary classification consisting of 500K positive data points randomly sampled from OpenWebMath (the seed corpus) and 500K negative data points randomly sampled from CommonCrawl. They train a fastText (Joulin, 2016) classifier on these data which they then use to extract samples from CommonCrawl as math content. All CommonCrawl domains for which over 10% of the existing web pages have been extracted are at this point understood to be math-related. URLs which are associated with these domains but which have yet to be collected are manually labeled as math content. The web pages hosted at these addresses are added to the seed corpus and the classifier is retrained. DEEPSEEK MATH performs 4 rounds in total resulting in the DEEPSEEK MATH Corpus, consisting of some 120B math tokens. They continuously train a partially converged 7B DEEPSEEK CODER-V1.5 model on a 500B token blend to attain the DEEPSEEK MATH model and achieve substantial improvement on several math tasks. In contrast, MIND proposes a simple alternative for generating high-quality math data that boosts the mathematical reasoning ability of LLM given access to a small seed corpus.

As the DEEPSEEK MATH dataset is not public, we replicate our previous blend, $\mathcal{D} = \{\mathcal{X} \cup \mathcal{R}_{pt}\}$, where $\mathcal{X} = \{\text{MIND-OWM-4B (conversations of all styles except the TWO STUDENTS one)} \cup \text{MIND-OWM-14B (TWO STUDENTS conversations)}\}$. We maintain a 2:1 ratio of \mathcal{X} and \mathcal{R}_{pt} in the training blend. Similar to the approach of DEEPSEEK MATH, we take a converged DEEPSEEK CODER-V1.5 model as \mathcal{C} — the unconverged model weights are unpublished as far as we are aware — and convert the model weights to a format compatible with Megatron-LM, which serves as our training framework, before continuously training for 500B tokens. We use a cosine learning rate schedule with a 19B token linear ramp-up, a maximum learning rate of $3e-4$, and a minimum learning rate of $3e-6$, and we anneal the learning rate over 500B tokens. We use Adam with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.95$, a weight decay of 0.1, a gradient clipping threshold of 1.0, a sequence length of 4096, and a global batch size of 2304 sequences.

Dataset	Tokens	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-Math	Avg-All
DEEPSEEK MATH (Zhihong Shao, 2024)		59.29	4.37	55.41	54.98	55.94	39.69	43.64
MIND-OWM-4B/14B [Combinations*]	500B	57.32	2.36	51.95	56.54	59.16	37.21	43.84

Table 12.21: **DEEPSEEK MATH vs All Synthetic Conversations.** A model trained on conversations generated by MIND from a small seed corpus can achieve math accuracy comparable to the DEEPSEEK MATH model trained on 120B unique tokens.

From Table 12.21, we can see that a model trained on conversations which MIND generated given a small seed corpus can attain math accuracies comparable to the DEEPSEEK MATH model with access to 120B unique math tokens in its continuous training blend. In fact, we outperform DEEPSEEK MATH in MMLU and general reasoning tasks, reaching higher average accuracy across all tasks. This underscores the quality of MIND generated conversations and signifies the efficacy of MIND in improving mathematical reasoning ability of LLM when the underlying raw data is limited.

In contrast to our prior \mathcal{C} , DEEPSEEK MATH-7B LLM is a strong math baseline that has been specifically designed for addressing mathematical reasoning ability and surpasses Azerbayev et al. (2023b), Team et al. (2024), Jiang et al. (2023), Lewkowycz et al. (2022a), Javaheripi et al. (2023), Dubey et al. (2024) [8B] base models on diverse math tasks. To evaluate the effectiveness of MIND with stronger pretrained model, we perform an additional experiment, similar to our training setup in Section 5.3.1 using $\mathcal{C} = \text{DEEPSEEK MATH-7B}$. Specifically, we have continuously trained the \mathcal{C} on 500B tokens maintaining a 2:1 ratio of math (\mathcal{R}) and 13 CC (\mathcal{R}_{pt}) dataset where the total blend is $\mathcal{D} = \{\mathcal{R} \cup \mathcal{R}_{pt}\}$. We conduct two experiments by alternating \mathcal{R} with raw (OWM-14B) and \mathcal{X} .

Dataset	Tokens	GSM8K	MATH	MMLU-STEM	MMLU	GENERAL REASONING (Avg)	Avg-Math	Avg-All
OWM-14B		39.42	1.59	49.92	52.87	55.47	30.31	37.34
MIND-OWM [ALL CONVERSATIONS]	500B	57.32	2.36	51.95	56.54	59.16	37.21	43.84

Table 12.22: **Training DEEPSEEK MATH-7B with Raw Data vs All Synthetic Dialogues.** A strong pretrained LLM continuously trained on conversations generated by MIND provides significant boost in math accuracy than the same model trained on raw data—showing the effectiveness of MIND regardless of pretraining model quality.

As shown in Table 12.22, model trained on MIND-OWM data shows consistent improvement over model trained on raw data—resulting in 17.90% gain on GSM8K, 6.90% average improvement across math tasks and 3.43% average improvement across ten general reasoning tasks. These results further solidifies the effectiveness of MIND regardless of the quality of the pretrained model.

12.10.5 Conversations on Code Tasks

Unlike raw data, conversations tend to break down the context into sub-context and participants exchange their reasoning about the sub-context in a single turn. This feature is particularly useful for mathematical or logical reasoning which require step-by-step reasoning. However, this structure might hurt performance of LLM in domains where sequence of context needs to be preserved such as in codes. To further investigate the impact of conversational data on the coding capabilities of LLM, we conduct an evaluation of models trained on both raw and synthetic data across four established coding benchmarks: HumanEval (Chen et al., 2021b), MBPP (Austin et al., 2021), HumanEval+, and MBPP+ (Liu et al., 2024a). These benchmarks are specifically designed to assess the model’s ability to generate functional code in response to given prompts.

Our results, as presented in Table 12.23, demonstrate that conversational synthetic data does not enhance coding performance. This is largely due to the way conversations tend to fragment code, wrapping it in natural language and thereby obscuring the intended sequence and logic inherent in programming tasks. Consequently, while conversations may be effective in contexts that benefit from collaborative reasoning, they are not suited for preserving the integrity of code, leading to diminished performance in coding benchmarks.

Interestingly, we also observe that rephrasing, which resembles raw data more closely in structure, further degrades coding accuracy. Our qualitative analysis of the rephrased documents reveals that the conversation generator (LLAMA3-70B-INSTRUCT) often re-

Dataset	Style	HumanEval	HumanEval+	MBPP (Sanitized)	MBPP+	Avg-All
OWM-4B	Raw	12.20	10.98	23.74	0.00	11.73
Rephrase-OWM-4B	Rephrase	5.49	4.27	20.23	0.53	7.63
MIND-OWM-4B	TWO PROFESSORS	8.54	4.88	20.62	0.00	8.51
	TEACHER STUDENT	13.41	9.76	26.46	0.26	12.47
	TWO STUDENTS	10.37	7.93	26.07	0.26	11.16
	LAYMAN KNOWALL	10.37	8.54	26.46	0.79	11.54
	DEBATE	11.59	9.15	24.90	0.26	11.48
	INTERVIEW	7.32	4.88	23.35	0.26	8.95
	PROBLEM SOLVING	9.76	9.15	24.51	0.26	10.92
	LONGEST CONVERSATION	9.15	7.32	28.40	0.53	11.35
	ALL CONVERSATIONS	12.20	9.15	28.02	0.53	12.48
OWM-4B+MIND-OWM-4B [1:1]	Combination	13.41	10.98	23.35	0.00	11.94
OWM-4B+MIND-OWM-4B [Concat]		10.37	7.93	31.52	0.00	12.46

Table 12.23: **Results of 7B LLM on Code Tasks:** Conversations and rephrases are ineffective for code benchmarks.

moves critical elements such as important tags and code indentation, which are essential for comprehending and executing code correctly. This finding underscores the necessity of designing code-specific prompts that retain the structural components vital for coding tasks.

Answer Questions from Image

Hi! Thanks for your help!
In this HIT, you are going to answer questions about images!

- For each question, **start typing your answer** in the textbox right to it. If you think the question is incorrect/unanswerable, please type "the question itself is problematic" and **provide an explanation** for choosing this option.
- To unlock the task, you need to **answer some questions correctly based on the instructions**. So, read the instructions, examples, and FAQs carefully!
- The answers are usually short, about **1-5 words**.

P.S. You will receive bonus if you can answer more than 8 questions correctly! So try to do your best! :) Good luck!

What is this HIT about? +-
Frequently Asked Questions (FAQ) +-
Examples +-
We are targeting questions that are needed to be grounded to image before answering them. Note that, before answering the question, you need to make sure all objects, relations and attributes mentioned in the question are present in the image. If not, the question is unanswerable i.e., "the question itself is problematic".

Example 1 [Question text is consistent with the image]: +-
Example 2 [Question text is NOT consistent with the image]: +-
Example 3 [Question text contains STAR RELATION]: +-
Example 4 [Question text contains CHAIN RELATION]: +-
Submit

Look at the examples above to get some hints about the task!

HINT: There are questions that are problematic themselves. Please read the Instructions carefully to understand their features. We REALLY need your help distinguishing them from the rest. We have also provided **Structured Representation of Question** (more information in FAQ) with the plain question text to make the question easier to read. And if you select any answers instead of the "the question itself is problematic" option for those questions, you will **FAIL!**

REMEMBER:

- ALL ANSWERS MUST BE INSIDE THE DROPDOWN LIST.
- ALL OBJECTS, THEIR ATTRIBUTES, AND THE RELATIONS AMONG THEM MENTIONED IN THE QUESTION TEXT MUST BE PRESENT IN THE IMAGE.

Answer these questions correctly using the information above to unlock the task!

Given the information above, which properties can make a question problematic? (Select options that are relevant) [See Instructions]

- Question text has an attribute for an object that is incorrect according to the image.
- Question text describes objects with their attributes and relations.
- Question text has an object that is not present in the image.
- Question text has two objects who/which share a relation that is not true according to the image.
- Question text is asking about an attribute that is visible in the image.

Do the 'no' and 'the question itself is problematic' options have similar meaning? [See Instructions]

- Yes
- No

What is the structure of STAR relation in a question text? [See FAQ]

- object1 -- relation1 -- object2, relation2 -- object3, and relation3 -- object4
- object1 -- relation1 -- object2 -- relation2 -- object3


What are examples of CHAIN relation? [See FAQ & Examples]

- What is the material property of the poll standing next to the road to the left of the car?
- What are the man wearing red shirt, standing on the table and holding a beer doing in the picture?
- Why is the bus to the right of the road, and to the left of the street light waiting?
- How many red books on the shelf to the left of the woman are new?

Submit

Image Question

Please click on image to expand



What are the black cat and the person wearing the black shoe and located to the left of the red ball and open door doing in the image?

Structured Representation of Question

Your Answer:

Thank you for completing our task!

- Please let us know if you faced any issues/confusion while solving this task in the **Optional Feedback** section.
- Please suggest us how we can improve. Your feedback is very valuable to us!

Have a good day!

Which question did you find easiest to answer?

1 2 3 4 5 6 7 8 9 10

Which question did you find most difficult to answer?

1 2 3 4 5 6 7 8 9 10

Optional Feedback:

Did you find the **Structured Representation of Question** useful?

Submit

Figure 12.8: Amazon Mechanical Turk interfaces used for Qualification Test to choose the right workers for human accuracy assessment on VISREAS task. We study the workers by deploying two tasks. In the first task, we ask the workers to read the instructions carefully (**Top left**) and answer some multiple-choice questions (**Top right**). After passing this task, ten questions with images will be presented and the final task would be to choose the right answer from the answer dropdown list (**Bottom right**). We choose the workers for the final evaluation who have correctly predicted more than seven answers out of ten questions.

Image

Please click on image to expand



Question

Are the birds, the truck and the car found parked under or parked on road in the image?

Your Answer:

Thank you for completing our task!

- Please let us know if you faced any issues/confusion while solving this task in the **Optional Feedback** section.
- Please suggest us how we can improve. Your feedback is very valuable to us!

Have a good day!

Complexity:

On a scale of 1 to 5, how hard was it to find the required answer in the image (1 Very Easy - 5 Very Hard)?

1 2 3 4 5

Did you find the question problematic while answering?

Optional Feedback:

Submit

(a)

Did you find the question problematic while answering?

- Did you find any ambiguity in the options while answering? For example synonymous words like apartment or houses.
- Yes No
- Have you found multiple possible answers in the image, and you have selected one amongst them?
- Yes No
- Do you think there are more than one objects in the image that are possibly addressed by the question but you were having trouble figuring out which one the question focused on?
- Yes No

Can you use your best effort to modify how the head object was described in the question to make it unambiguous? Please make minimal/necessary modifications while preserving the original meaning of the question as much as possible.

Did you find the question problematic while answering?

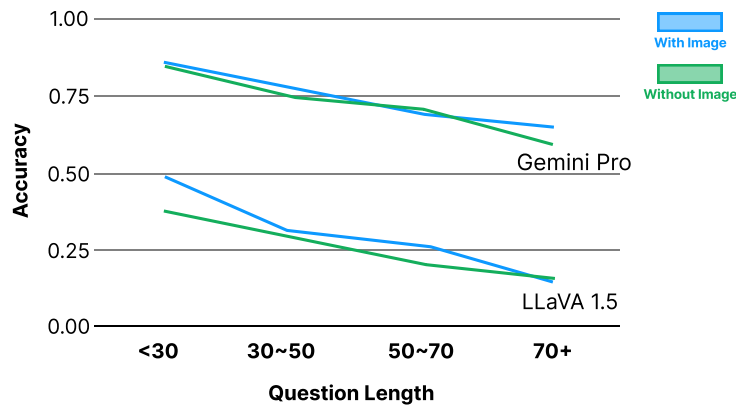
You understand the question. But you find the relation descriptions in this question problematic:

- No, it is perfect
- Yes, it is redundant (not needed at all as the object is easily locatable in the image from its name)
- Yes, it is too lengthy (the head obj has to be located via another related object, but the relation descriptions don't have to be multi-step)

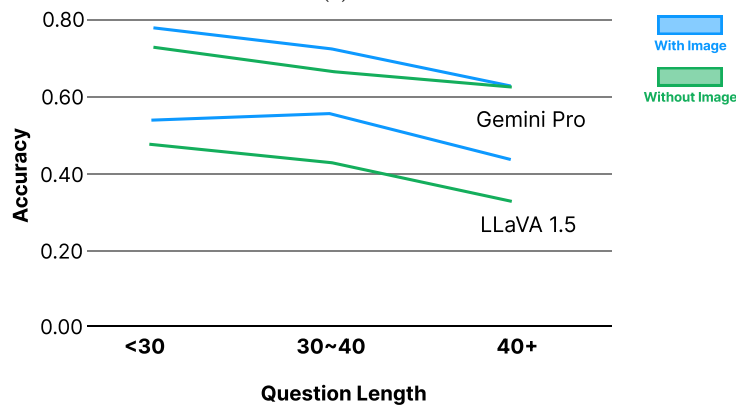
Can you please rewrite the question by simplifying the relation descriptions? (You must keep at least one relation. Otherwise, please select the "redundant" choice above.)

(b)

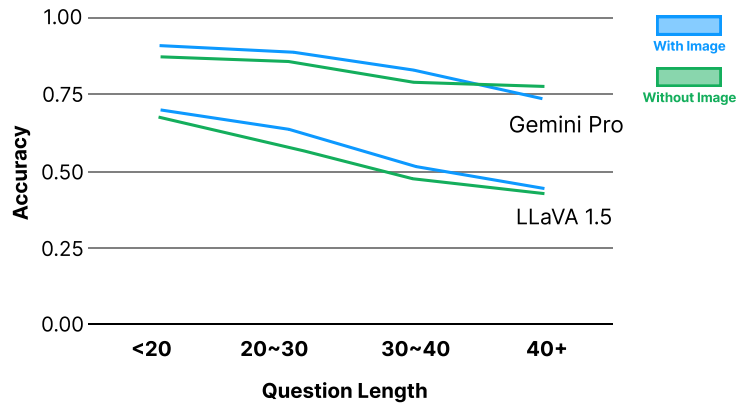
Figure 12.9: Amazon Mechanical Turk interfaces for human accuracy assessment on VIS-REAS task using the qualified workers. (a) For each HIT, we provide an image and a question that needs to be answered from a dropdown list of 2013 entries. In addition, we ask for rating the complexity and structural soundness of the query and further look for details if any Turker finds the question problematic. (b) To investigate what type of problem the question possesses, we ask for further details from the workers and even encourage them to rewrite the query to remove the problem they faced while answering the query.



(a) GSM8K



(b) SVAMP



(c) ASDIV

Figure 12.10: Accuracy by question length across three mathematical reasoning tasks. In the cases of ASDIV and SVAMP, accuracy is notably higher when utilizing images for longer and more intricate questions compared to scenarios without images. However, in the context of more complex questions, such as those found in GSM8K, the limitations of the VLM become apparent. In this scenario, the inability to generate effective HTML results in erroneous image generation, consequently leading to decreased accuracy, particularly with longer questions.

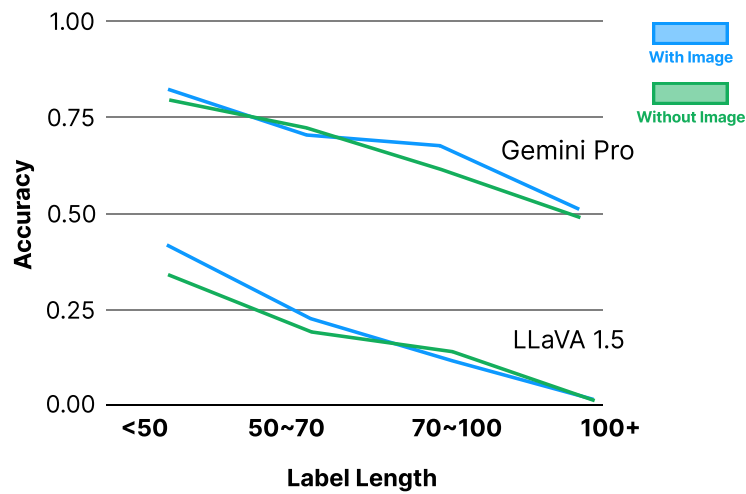


Figure 12.11: GSM8K accuracy by chain-of-thought length. Similar to the findings in [Figure 12.10](#), image representations for complex questions are not efficient and structured. Therefore, the inclusion of images does not enhance the representation of questions that demand longer chains of thought.

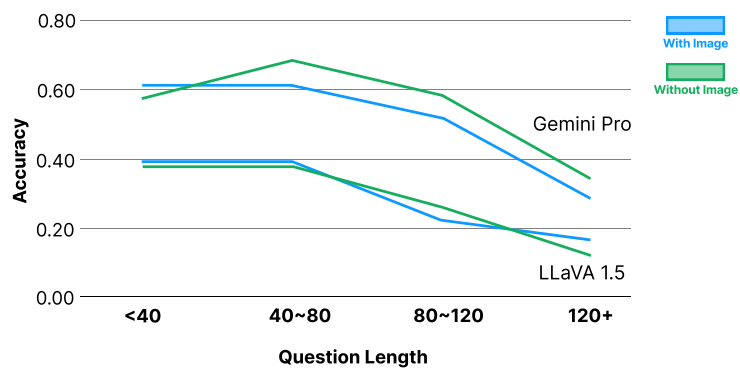


Figure 12.12: Accuracy by Question Length for a subset of BIG-Bench-Hard benchmark. Incorporating images helps when the corresponding question is simpler and shorter and when the questions are more complex.



Figure 12.13: Number of Instances from each subtask impacted by Image. Here ‘Image Hurts’ represents instances that achieved correct answers without image and got incorrect with image. Similarly ‘Image Improves’ shows data points getting the correct answers with image and getting incorrect without image.

12.11 Dialogue Examples

RAW TEXT

CAT Quantitative Aptitude Questions | CAT Ratios, Mixtures, Alligations and Averages Questions

CAT Questions | Mixtures and Alligations | Mixtures - Range

The question is from CAT Ratio, Mixtures and Averages. Given the details about two alloys, we need to determine the possible values the weight of an ally can take. CAT exam is known to test on basics rather than high funda ideas. A range of CAT questions can be asked from Ratios and Proportions, Mixtures, Alligations and Averages. Make sure you master the topics. 2IIMs CAT questions bank provides you with CAT questions that can help you gear for CAT Exam CAT 2020.

Question 16: 100 kgs of an alloy of tin and lead in the ratio 1:3 is mixed with x kgs of an alloy of tin and lead in the ratio 3:2. If the overall alloy should contain between 40% and 50% tin, what is the range of values x can take?

1. $100 \text{ kgs} \leq x \leq 200 \text{ kgs}$
2. $80 \text{ kgs} \leq x \leq 240 \text{ kgs}$
3. $110 \text{ kgs} \leq x \leq 220 \text{ kgs}$
4. $75 \text{ kgs} \leq x \leq 250 \text{ kgs}$

Best CAT Coaching in Chennai

CAT Coaching in Chennai - CAT 2020 Online Batches Available Now!

SYNTHETIC DIALOGUE [TWO PROFESSORS]

Professor 1: Good morning, colleague. I'd like to discuss a question from CAT Ratio, Mixtures, and Averages. It's about two alloys of tin and lead, and we need to determine the possible values of the weight of an alloy.

Professor 2: Ah, yes. I'm familiar with this type of question. It's a classic example of Mixtures and Alligations. Please go ahead and share the details of the question.

Professor 1: We have 100 kgs of an alloy of tin and lead in the ratio 1:3, which means it has 25 kgs of tin and 75 kgs of lead. This alloy is mixed with x kgs of another alloy of tin and lead in the ratio 3:2. The overall alloy should contain between 40% and 50% tin. We need to find the range of values x can take.

Professor 2: That's a great problem. We can start by analyzing the second alloy. If it has a ratio of 3:2, that means it has $\frac{3x}{5}$ kgs of tin and $\frac{2x}{5}$ kgs of lead, with a total weight of x kgs.

Professor 1: Exactly. Now, when we mix these two alloys, how does the mixture change? We need to find the overall percentage of tin in the resulting alloy.

Professor 2: Right. Let's think about it. If the overall alloy should contain between 40% and 50% tin, what does that imply about the range of values x can take?

Professor 1: That's the key question. If we can determine the range of x , we can answer the question. Do you think it's possible to narrow down the options to one of the given choices?

Professor 2: Yes, I think so. Let's analyze the options. We have 1. $100 \text{ kgs} \leq x \leq 200 \text{ kgs}$, 2. $80 \text{ kgs} \leq x \leq 240 \text{ kgs}$, 3. $110 \text{ kgs} \leq x \leq 220 \text{ kgs}$, and 4. $75 \text{ kgs} \leq x \leq 250 \text{ kgs}$. Which one do you think is the correct answer?

Professor 1: Well, after analyzing the problem, I think the correct answer is option 2: $80 \text{ kgs} \leq x \leq 240 \text{ kgs}$. What do you think?

Professor 2: I agree with you. The correct answer is indeed option 2. Well done! This question is a great example of how CAT exam tests our understanding of basics rather than high-funda ideas.

RAW TEXT

?

Free Version

Moderate

Interpretation of Equilibrium from Graph of Concentration vs Time

CHEM-NWU9LE

Consider the graph below of a chemical reaction observing the concentration of two chemical species with time.

Which species is the reactant? \$\$

Which species is the product?

\$\$

At what region does the solution appear to be in equilibrium? \$\$

What is the K_c for the reaction?

SYNTHETIC DIALOGUE [TEACHER STUDENT]

Student: Hi, teacher. I'm having trouble understanding this graph of a chemical reaction. Can you help me?

Teacher: Of course! What's your first question about the graph?

Student: Okay, so which species is the reactant?

Teacher: Let's take a look at the graph. The species whose concentration is decreasing with time is the reactant. Can you identify which one that is?

Student: Ah, I think it's... (looks at the graph) ...the one whose concentration starts high and decreases.

Teacher: That's correct! The reactant is the species whose concentration starts high and decreases with time. Now, what's your next question?

Student: Which species is the product?

Teacher: Good question! The product is the species whose concentration starts low and increases with time. Can you identify which one that is?

Student: Yeah, I think it's... (looks at the graph) ...the one whose concentration starts low and increases.

Teacher: That's right! The product is the species whose concentration starts low and increases with time. Now, what's your next question?

Student: At what region does the solution appear to be in equilibrium?

Teacher: Look at the graph again. At what point does the concentration of both species seem to be constant, not changing with time?

Student: Hmm... I think it's... (looks at the graph) ...after a certain time, when the lines are flat.

Teacher: Exactly! The solution appears to be in equilibrium when the concentration of both species is constant, which is after a certain time when the lines are flat. Now, what's your next question?

Student: What is the K_c for the reaction?

Teacher: Unfortunately, the graph doesn't provide enough information to calculate the K_c value. We would need more data or information about the reaction to determine the equilibrium constant, K_c .

Student: Okay, I think I understand now. Thank you for your help!

RAW TEXT

What's the probability that: The person is not age 18 to 34 and does not own an iPhone?

The presidential election is coming.

Five survey companies (A, B, C, D, and E) are doing survey to forecast whether or not the Republican candidate will win the election.

Each company randomly selects a sample size between 1000 and 1500 people.

All of these five companies interview people over the phone during Tuesday and Wednesday.

The interviewee will be asked if he or she is 18 years old or above and U.S. citizen who are registered to vote. If yes, the interviewee will be further asked: will you vote for the Republican candidate?

On Thursday morning, these five companies announce their survey sample and results at the same time on the newspapers.

The results show that $a\%$ (from A), $b\%$ (from B), $c\%$ (from C), $d\%$ (from D), and $e\%$ (from E) will support the Republican candidate.

The margin of error is plus/minus 3% for all results.

Suppose that $c > a > d > e > b$. When you see these results from the newspapers, can you exactly identify which result(s) is (are) not reliable and not accurate?

That is, can you identify which estimation interval(s) does (do) not include the true population proportion?

If you can, explain why you can, if no, explain why you cannot and what information you need to identify.

Discuss and explain your reasons. You must provide your statistical analysis and reasons.

SYNTHETIC DIALOGUE [TWO STUDENTS]

****Turn 1****

Alex: Hey Ben, have you started working on our assignment about the presidential election survey?

Ben: Yeah, I've been going over the context. It seems like we need to identify which survey result is not reliable and accurate.

Alex: Exactly! Let's break it down. We have five survey companies, A, B, C, D, and E, each with a sample size between 1000 and 1500 people.

Ben: Right, and they all have a margin of error of plus/minus 3%.

****Turn 2****

Alex: Okay, so we're given that $c > a > d > e > b$. What does this tell us?

Ben: It tells us the order of the proportions of people who will support the Republican candidate according to each survey company.

Alex: That's right. And since the margin of error is the same for all companies, we can assume that the true population proportion should lie within a similar range for all companies.

****Turn 3****

Ben: But wait, if the true population proportion lies within a similar range, then shouldn't all the estimation intervals overlap?

Alex: Exactly! If the estimation intervals don't overlap, it means that at least one of them is not reliable and accurate.

Ben: So, we need to check if the intervals overlap. How do we do that?

****Turn 4****

Alex: Let's calculate the estimation intervals for each company. For company A, it's $a\% \pm 3\%$, and similarly for the others.

Ben: Okay, got it. So, we have $a\% \pm 3\%$, $b\% \pm 3\%$, $c\% \pm 3\%$, $d\% \pm 3\%$, and $e\% \pm 3\%$.

Alex: Now, let's check if these intervals overlap. Since $c > a > d > e > b$, the interval for company C should be the highest, and the interval for company B should be the lowest.

****Turn 5****

Ben: I see what you're getting at. If the intervals overlap, then the highest value of the lowest interval (company B) should be greater than or equal to the lowest value of the highest interval (company C).

Alex: That's right! If that's not the case, then we know that at least one of the intervals is not reliable and accurate.

Ben: Let's plug in some numbers and see if we can identify which one is not reliable.

****Turn 6****

Alex: Alright, let's say the results are $a = 45\%$, $b = 38\%$, $c = 50\%$, $d = 42\%$, and $e = 40\%$. Can we identify which one is not reliable?

Ben: Let's calculate the estimation intervals... (calculates)...and check if they overlap.

Alex: (checks the calculations) Ah, look at that! The interval for company B ($38\% \pm 3\%$) doesn't overlap with the interval for company C ($50\% \pm 3\%$).

Ben: That means company B's result is not reliable and accurate!

****Turn 7****

Ben: But why can't we identify the other unreliable results, if any?

Alex: That's because we only have the relative order of the proportions, not the exact values. We need more information to identify other unreliable results, if any.

Ben: I see. So, we can only identify company B's result as not reliable and accurate, but we can't say for sure about the others.

Alex: That's correct. We've done our best with the given information.

RAW TEXT

Decimal to Percent Calculator.

The calculator will convert decimal into percent, with steps shown.

If the calculator did not compute something or you have identified an error, or you have a suggestion/feedback, please write it in the comments below.

Convert 1.4 into decimal. ## Solution From the converting decimal to percent note, we know that to obtain a percent, we need to multiply a decimal by 100.

But multiplying by 100 is equivalent to moving a decimal point 2 positions to the right.

SYNTHETIC DIALOGUE [LAYMAN KNOWALL]

****Turn 1****

Layman: Hi, I'm not sure what this calculator does. Can you explain it to me?

Me: This calculator converts decimals into percents. It shows the steps to do the conversion.

****Turn 2****

Layman: Okay, got it. So what's the problem we're trying to solve?

Me: We're trying to convert 1.4 into a percent.

****Turn 3****

Layman: How do we do that? Is there a formula or something?

Me: Yes, there is. According to the note on converting decimal to percent, we need to multiply the decimal by 100 to obtain a percent.

****Turn 4****

Layman: Why do we need to multiply by 100? What's the logic behind it?

Me: Well, I'm not sure about the logic, but the note says that's what we need to do. But I can tell you that multiplying by 100 is equivalent to moving the decimal point 2 positions to the right.

****Turn 5****

Layman: What do you mean by moving the decimal point 2 positions to the right? Can you explain that?

Me: Sure. When you multiply a number by 100, the decimal point shifts 2 places to the right. For example, if you have 1.4 and you multiply it by 100, the decimal point will move 2 places to the right, making it 140.

****Turn 6****

Layman: Ah, I think I understand now. So, if we multiply 1.4 by 100, what do we get?

Me: That's right! If we multiply 1.4 by 100, we get 140. And that's the equivalent percent value of the decimal 1.4.

RAW TEXT

Try to finish the derivation and see how it applies to the sum you're trying to compute. Sorry if this is hard to understand.

7. Jun 14, 2009

jgens

Cyosis, why would he/she use the sum for the first n natural numbers when his sum asks for odd natural numbers only?

8. Jun 14, 2009

Cyosis

If you know the sum for the first n natural numbers you can quickly solve the sum for the first odd numbers. Had he known it, it would have been the easiest way to calculate the sum he's interested in.

9. Jun 14, 2009

Manchot

The story is that a six-year-old Gauss was told by his teacher to add the numbers 1 to 100 as a form of busywork. In seconds, he came up with the answer, much to his teacher's surprise. What he did was notice that since $1+100=101$, $2+99=101$, $3+98=101$, etc., the sum becomes a simple multiplication: there are 50 of the above pairs, so the sum is just $50 \cdot 101 = 5050$. Anyway, that's the intuition behind the identity that Cyosis posted. You can pretty much use the exact same intuition.

10. Jun 14, 2009

gabrielh

Okay thanks, I get it now.

11. Jun 14, 2009

SYNTHETIC DIALOGUE [DEBATE]

jgens: I don't understand why Cyosis suggested using the sum of the first n natural numbers to solve the problem. The original problem asks for the sum of the first odd natural numbers, not all natural numbers.

Cyosis: I see what jgens is saying, but I think my approach is still valid. If you know the sum of the first n natural numbers, you can easily derive the sum of the first odd natural numbers. It's actually the easiest way to calculate the sum.

Manchot: I think I can help clarify things. The story about Gauss adding numbers from 1 to 100 is relevant here. He noticed that $1+100=101$, $2+99=101$, and so on, which makes the sum a simple multiplication problem. There are 50 such pairs, so the sum is $50 \cdot 101 = 5050$. This is the intuition behind Cyosis' suggestion.

jgens: I see what Manchot is saying, but I'm still not convinced. How does this story about Gauss relate to the sum of odd natural numbers?

Cyosis: Well, jgens, the same intuition applies here. We can use a similar pairing method to calculate the sum of odd natural numbers. It's not a direct application, but the principle is the same.

gabrielh: I think I'm starting to get it now. So, we can use the sum of the first n natural numbers as a stepping stone to calculate the sum of the first odd natural numbers, and the Gauss story provides the intuition behind this approach.

Manchot: Exactly, gabrielh! That's the idea. By understanding the sum of the first n natural numbers, we can derive the sum of the first odd natural numbers using a similar method.

Cyosis: I'm glad we could clarify things. The key is to recognize that the sum of the first n natural numbers is a more general problem that can be used to solve the more specific problem of the sum of the first odd natural numbers.

RAW TEXT

The Cryptographic Lens

Shafi Goldwasser - Colloquium Lecture - POSTPONED!

Tuesday, 21.10.2014, 14:30

Room 337-8 Taub Bld.

...

ceClub: A Centralized Zero-Queue Network Architecture

Jonathan Perry (MIT, CSAIL)

Monday, 06.10.2014, 11:00

EE Meyer Building 861

Current datacenter networks inherit the principles that went into the design of the Internet, where packet transmission and path selection decisions are distributed among the endpoints and routers. Instead, we propose that each sender should delegate control—to a centralized arbiter—of when each packet should be transmitted and what path it should follow. Fastpass is a datacenter network architecture built using this principle. Fastpass incorporates two fast algorit...

Parameterized Automata Constructions and Their Applications

Ran Ben-Basat

Monday, 22.09.2014, 15:30

Taub 701

Parameterization is a useful tool for handling NP-hard problems in the real world. It aims to reduce the running times of algorithms for such problems, by confining the combinatorial explosion to some parameter k . As this parameter is often significantly smaller than the input size, it allows to develop practical algorithms for non-trivial classes of instances for these problems. In this talk we present a novel framework for developing parameterized algorithms, using construction...

Observing the Observers: Social Context Analysis Using Computer Vision

Meir Cohen

Wednesday, 10.09.2014, 12:00

Taub 601

It is quite common that multiple human observers attend to a single point of interest. Mutual awareness activity (MAWA) refers to the dynamic of this social phenomena. A peak of a MAWA is known as a mutual awareness event (MAWE) and can be interpreted as a buzz event, which draws the attention of many observers.

A preferred way to monitor those social phenomenon is with a camera that captures the human observers while they observe the activity in the scene.

Our work studies the ...

SYNTHETIC DIALOGUE [INTERVIEW]

Subject Matter Expert: Unfortunately, the lecture was postponed, so we don't have any information about the specific topic she was going to cover. However, the title of the lecture was The Cryptographic Lens, which suggests that it might have been related to cryptography and its applications.

Interviewer: That's a shame it was postponed. Moving on, let's talk about the ceClub lecture by Jonathan Perry from MIT, CSAIL. Can you summarize the main idea behind his proposed datacenter network architecture, Fastpass?

Subject Matter Expert: Yes, certainly. Jonathan Perry proposed a centralized zero-queue network architecture called Fastpass, which is a departure from the traditional distributed approach used in Internet design. In Fastpass, each sender delegates control to a centralized arbiter, which decides when each packet should be transmitted and what path it should follow. This approach aims to improve the efficiency of datacenter networks.

Interviewer: That's fascinating. What can you tell me about the talk by Ran Ben-Basat on parameterized automata constructions and their applications?

Subject Matter Expert: Ran Ben-Basat discussed the concept of parameterization, which is a useful tool for handling NP-hard problems in the real world. The idea is to reduce the running times of algorithms for these problems by confining the combinatorial explosion to some parameter k . He presented a novel framework for developing parameterized algorithms using construction-based methods.

Interviewer: I see. Now, let's switch gears and talk about the lecture by Meir Cohen on social context analysis using computer vision. Can you explain what mutual awareness activity (MAWA) and mutual awareness event (MAWE) refer to in this context?

Subject Matter Expert: Yes. Mutual awareness activity (MAWA) refers to the dynamic of multiple human observers attending to a single point of interest. A peak of this activity is known as a mutual awareness event (MAWE), which can be interpreted as a buzz event that draws the attention of many observers. Meir Cohen's work focuses on monitoring and analyzing these social phenomena using computer vision, specifically by capturing the human observers with a camera while they observe the activity in the scene.

RAW TEXT

Note: At section $X = 12$ m, the speed is close to 1.5 m / s Thus for a return trip of $2 \times 20 \text{ m} = 40$ m to return to the profile $X = 14 \text{ m}$, the wave takes 27 s. Thus, the graph of Figure. 11 is located before the (eventual) reflected wave coming from the absorbing upstream boundary, which can disturb the sensors.

Comparative analysis of records

We have exploited some of the tests in a transversal manner, to identify particular behaviors of the surface waves:

- impact of the variation of the amplitude for several tests characterized by the same parameters but with roughnesses, or geometries either uniform, or linear
- impact of the change in the period
- impact of the variation of the initial amplitude
- comparison for the same test carried out in uniform or converging section
- impact of roughness variation

Impact of the variation of the amplitude for several tests with the same parameters

We have compared several tests characterized by D70, by Ab absorbing boundary conditions and for the same period T5. These are tests D70A5T5, D70A2T2 and D70A5T7 in RoAbUn, SmAbCo and SmAbUn conditions

Figure 16 Figure 17 Figure 18

- Figure 16: amplitudes variations for the 3 runs are increasing according to the slope, which is in accordance with the theory.
- Figure 17: the initial amplitudes of the waves generated by the wave maker are not always well respected: all the curves of the same diagram should start from the same point to $X = 0$
- Figure 18: there is a sharp change in amplitude at the arrival at the top of the slope.

SYNTHETIC DIALOGUE [PROBLEM SOLVING]

User: Hi, I’m analyzing some data on surface waves and I’d like to discuss some observations with you. Can you help me understand the impact of varying amplitude on surface waves?

AI Assistant: I’d be happy to help. From the context, I see that you have several tests with the same parameters but different amplitudes. Specifically, you have tests D70A5T5, D70A2T2, and D70A5T7 in RoAbUn, SmAbCo, and SmAbUn conditions. What would you like to know about these tests?

User: Yes, that’s correct. I’m looking at Figures 16, 17, and 18. Can you explain what’s happening in Figure 16?

AI Assistant: In Figure 16, the amplitudes of the waves are increasing according to the slope, which is consistent with theoretical expectations. This suggests that the amplitude of the wave is directly related to the slope of the surface.

User: That makes sense. But what about Figure 17? The initial amplitudes of the waves generated by the wave maker don’t seem to be consistent.

AI Assistant: That’s a good observation. Yes, the initial amplitudes of the waves don’t start from the same point at $X = 0$, which is unexpected. This could be an issue with the wave maker or the experimental setup.

User: Okay, got it. And what about Figure 18? There seems to be a sudden change in amplitude at the top of the slope.

AI Assistant: Yes, that’s correct. Figure 18 shows a sharp change in amplitude at the arrival at the top of the slope. This could be due to the interaction between the wave and the slope, causing a sudden increase or decrease in amplitude.

User: That’s helpful. So, what can we conclude about the impact of varying amplitude on surface waves?

AI Assistant: Based on these figures, it seems that the amplitude of the wave is directly related to the slope of the surface, and changes in amplitude can affect the behavior of the wave. However, we also need to consider the potential issues with the wave maker or experimental setup that may be affecting the results.

User: Okay, that’s a good summary. Thanks for your help in analyzing these figures!

12.12 Data Proportion across Blends

To better understand the data composition used in our reinforcement learning experiments, we report the proportion of each dataset in the six blending strategies introduced in Section 10.2. These proportions reflect how data is distributed across different sources depending on the specific blending paradigm: data source, question type, and data usefulness.

Data Name	Type	\mathcal{B}_{nd}	$\mathcal{B}_{mr\uparrow}$	$\mathcal{B}_{mcq\uparrow}$	$\mathcal{B}_{open\uparrow}$	$\mathcal{B}_{gpr\uparrow}$	\mathcal{B}_{score}	\mathcal{B}_{only_math}	\mathcal{B}_{only_gpr}
MMLU	MCQ	0.1696	0.0864	0.2251	0.1159	0.1678	0.1296		0.2542
Syn-QA	MCQ	0.3277	0.1670	0.4349	0.2241	0.3242	0.1731		0.4912
NATURAL REASONING	OPEN-ENDED	0.1699	0.0866	0.1149	0.2231	0.1680	0.1683		0.2546
NuminaMath	OPEN-ENDED	0.1484	0.2943	0.1004	0.1949	0.1516	0.2020	0.4460	
Persona-MATH	OPEN-ENDED	0.1699	0.3370	0.1149	0.2231	0.1736	0.1579	0.5105	
MATH	OPEN-ENDED	0.0145	0.0287	0.0098	0.0190	0.0148	0.1691	0.0435	

Table 12.24: Proportion of each dataset in different blends.

12.13 Token Efficiency Analysis

Token Efficiency in Correct Responses. Understanding not only whether a model answers correctly but also how efficiently it reasons is critical in real-world deployments, especially for reducing inference cost and latency. To this end, we analyze the token lengths of correct responses generated by models trained under different data blending strategies.

Table 12.25 presents the minimum, maximum, and mean number of tokens used in correct answers across two task types: General Purpose Reasoning (GPR) and Math. We compare three models: (1) $\mathcal{B}_{gpr\uparrow}$ (multi-domain training), (2) \mathcal{B}_{only_math} (math-only training), and (3) ORZ (a strong math-centric baseline model).

Across GPR tasks, $\mathcal{B}_{gpr\uparrow}$ produces the most concise correct responses, with a mean of 385 tokens—39.6% fewer than \mathcal{B}_{only_math} and 65.4% fewer than ORZ. This suggests that training with multi-domain corpora equips the model to reason more efficiently in less structured tasks, avoiding unnecessarily verbose responses.

Task Type	Model	Min	Max	Mean
GPR	$\mathcal{B}_{gpr\uparrow}$	83.20	2697.80	385.41
	\mathcal{B}_{only_math}	159.60	9594.00	638.57
	ORZ	223.00	8221.80	1114.60
Math	$\mathcal{B}_{gpr\uparrow}$	170.25	10130.00	622.00
	\mathcal{B}_{only_math}	201.75	11330.25	730.68
	ORZ	292.00	12917.00	1257.00

On math benchmarks, where detailed step-by-step derivations are essential, all models naturally generate longer outputs. However, $\mathcal{B}_{gpr\uparrow}$ still demonstrates adaptability, producing appropriately longer responses compared to GPR, while keeping the output concise relative to \mathcal{B}_{only_math} and ORZ. This behavior underscores the ability of multi-domain trained models to dynamically adjust their reasoning strategy and verbosity based on task requirements.

Table 12.25: Token length statistics (Min, Max, Mean) for correct responses across task types.

Interestingly, ORZ exhibits the longest response lengths across both GPR and math tasks. While this aligns with its design as a reasoning-heavy model, it also reflects less efficiency—potentially generating unnecessarily long chains of thought, particularly in domains outside its training focus.

In summary, the token efficiency analysis reveals that $\mathcal{B}_{gpr\uparrow}$ achieves a favorable trade-off between accuracy and brevity, tailoring its reasoning depth to the complexity of the task. This reinforces the value of diverse, multi-domain training in promoting adaptable and cost-efficient language models.

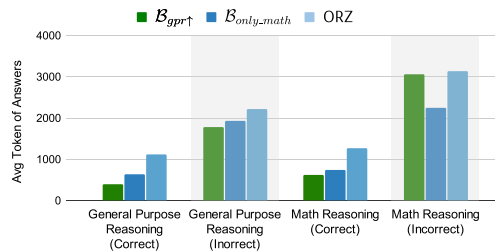


Figure 12.16: Average token lengths of correct and incorrect responses across general-purpose and math reasoning tasks for models trained on $\mathcal{B}_{gpr\uparrow}$, \mathcal{B}_{only_math} , and ORZ.

Thinking Long vs Thinking Accurate.

Recent studies such as DeepScaler (Luo et al., 2025a) have noted that incorrect answers often exhibit longer trajectories, leading to wasted computation and less efficient learning. Echoing this observation, we analyze the average token lengths of correct and incorrect responses for models trained on different blends: $\mathcal{B}_{gpr\uparrow}$, \mathcal{B}_{only_math} , and ORZ.

As shown in Figure 12.16, incorrect responses are consistently and substantially longer than correct ones—by $3.6\times$ on average. This pattern holds across both general-purpose and math reasoning tasks, suggesting that verbose reasoning does not guarantee correctness. In fact, longer responses often reflect the model’s uncertainty, overthinking, or repetitive CoT traces, rather than productive deduction.

12.14 Sub-category Accuracy Analysis

To further support our observation that multi-domain training improves general-purpose reasoning while remaining competitive on math tasks, we analyze the number of correct responses across sub-categories in MMLU-PRO and AGIEVAL. Figure 12.17 and Figure 12.18 show the count of correct answers produced by $\mathcal{B}_{gpr\uparrow}$ and \mathcal{B}_{only_math} across their respective sub-domains.

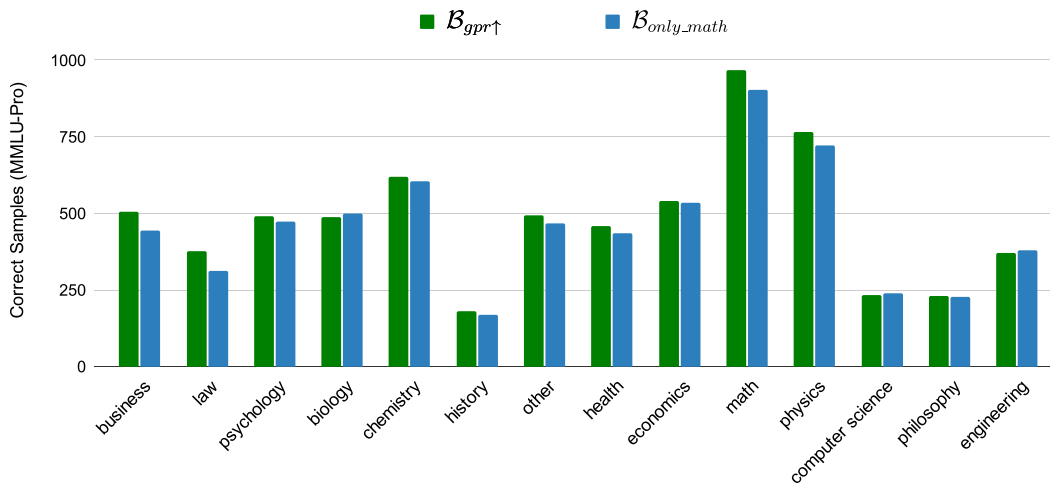


Figure 12.17: **Sub-category Accuracy Comparison across MMLU-PRO Domains.** The $\mathcal{B}_{gpr\uparrow}$ blend consistently outperforms \mathcal{B}_{only_math} in a wide range of non-math reasoning categories such as business, law, psychology, and economics. Surprisingly, it also slightly surpasses the math-specialized blend in the MMLU-PRO math category, highlighting the generalizability and versatility of multi-domain training.

On MMLU-PRO, $\mathcal{B}_{gpr\uparrow}$ consistently outperforms \mathcal{B}_{only_math} across non-math reasoning categories such as business, law, psychology, chemistry, and economics. Notably, it achieves relative improvements of +20.58% in law and +13.26% in business. Surprisingly, $\mathcal{B}_{gpr\uparrow}$ also performs better in the math category (+7.2%), despite not being trained exclusively on mathematical data. This may be attributed to the nature of MMLU-PRO’s math problems, which are college-level and benefit from a combination of symbolic and heuristic reasoning—skills reinforced through exposure to diverse domains.

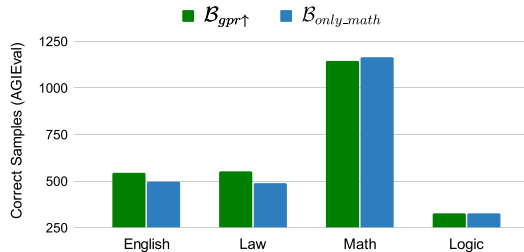


Figure 12.18: **Sub-category Accuracy Comparison across AGIEVAL.** While \mathcal{B}_{only_math} performs marginally better in the math, $\mathcal{B}_{gpr\uparrow}$ achieves stronger results in non-math domains.

In contrast, the AGIEVAL benchmark (shown in Figure 12.18) features Olympiad-level math questions that are more abstract and complex. Here, \mathcal{B}_{only_math} has a slight edge (+1.8%) in the math category, which aligns with its domain-specific training. However, $\mathcal{B}_{gpr\uparrow}$ demonstrates stronger performance in symbolic and language-heavy domains, showing +13.06% improvement in Law and +9.88% in English. Averaged across all non-math reasoning categories, $\mathcal{B}_{gpr\uparrow}$ achieves a +8.6% relative gain over \mathcal{B}_{only_math} , reinforcing its advantage in general-purpose and real-world reasoning tasks.

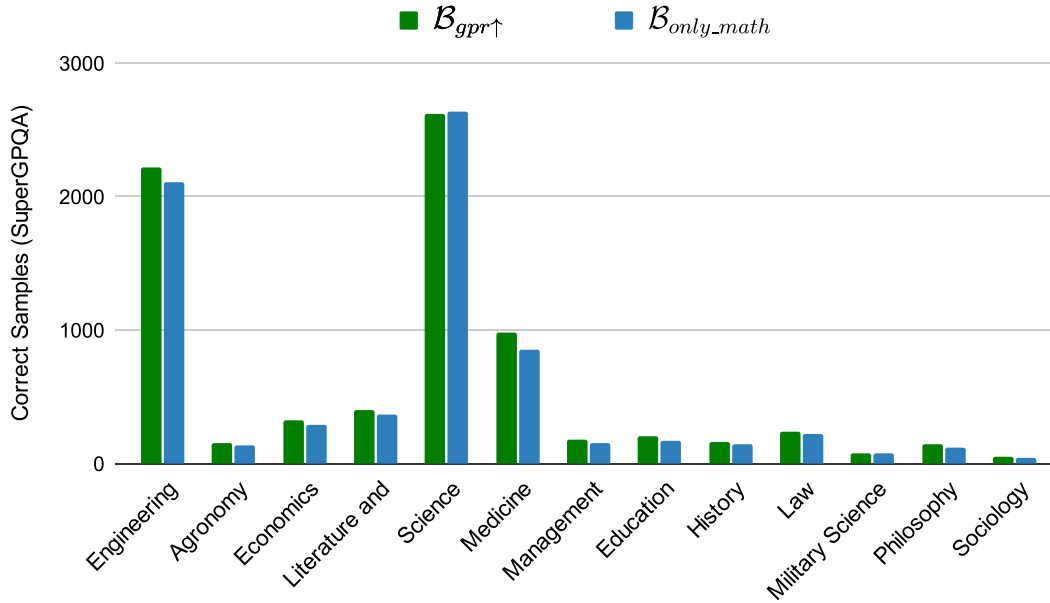


Figure 12.19: **Sub-category Accuracy Comparison across SUPERGPQA.** The $\mathcal{B}_{gpr\uparrow}$ blend consistently outperforms \mathcal{B}_{only_math} in a wide range of non-math reasoning categories except the science category which consists of fields like mathematics, physics, astronomy, chemistry etc.—highlighting the generalizability and versatility of multi-domain training.

A similar trend is observed in the SUPERGPQA sub-category analysis shown in Figure 12.19. $\mathcal{B}_{gpr\uparrow}$ significantly outperforms \mathcal{B}_{only_math} across nearly all categories—especially in engineering, agronomy, economics, education, law, and philosophy. The only exception is the “Science” category, which includes math-heavy disciplines like physics, chemistry, and astronomy, where both blends perform comparably. This further highlights that multi-domain training with $\mathcal{B}_{gpr\uparrow}$ enhances reasoning across a broad spectrum of fields, achieving strong generalization even in real-world, professional domains that fall outside traditional math tasks.

12.15 Relative Importance of Structure and Context: Evidence from Ablations

A natural question is whether *structure* (e.g., explicit reasoning flow, conversational dynamics, reflection–refine cycles) is as important as *context* (e.g., domain, topic, or raw content) in driving gains. The six attributes above are not orthogonal; their relative importance depends on the target task and the goal (narrow peak performance vs. broad, scalable improvement).

This section summarizes empirical evidence from our work and a controlled long-thought ablation to clarify which factors lead to observed differences.

Structure versus context. Evidence from this thesis suggests that **structure is at least as important as context** for complex reasoning. In MIND (chapter 5), raw web text from OpenWebMath is transformed into structured *conversations* with knowledge gaps between participants; the same underlying context (math documents) is retained, but the format shifts from plain text to multi-turn dialogue. Models trained on MIND-generated data consistently outperform those trained on raw or rephrased data, with gains of 4.78%–12.82% on GSM8K and 0.54%–1.28% on MATH (see Table 5.1 in chapter 5). The disparity is attributed to the fact that rephrasing preserves syntactic meaning but does not add semantic or pragmatic variation, whereas “the structured and interactive nature of conversations facilitates a more nuanced understanding of the problem space” (see chapter 5). Thus, for mathematical reasoning, *how* the content is organized (conversational dynamics as inner monologue) matters as much as *what* the content is. At the same time, MIND excels on GSM8K-style and general math benchmarks but does not by itself provide the multiple reflection and refine cycles that benefit the hardest math problems (e.g., MATH-500, competition-level). For those, explicit *structured reasoning flow*—hypothesis generation, solution attempts, verification, and refinement—appears necessary.

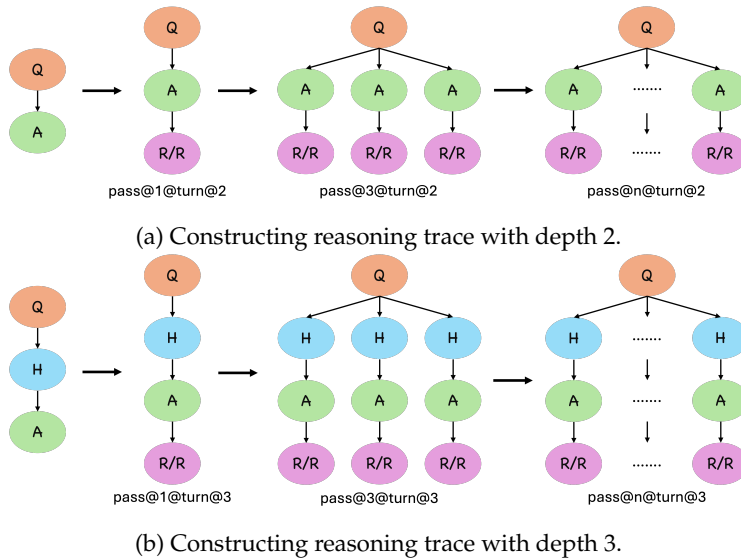


Figure 12.20: Guided Multi-Step Reasoning Expansion.

Impact of reasoning components: long-thought ablation. To isolate the impact of reasoning *components* (e.g., hypothesis, answer, reflection, refine) within long-thought-style data, we conducted an ablation following a long-thought pretraining plan: varying pipelines from simple question-to-solution to multi-step pipelines that include hypothesis generation, multiple solution attempts, and reflect-and-refine stages.¹ For this, we use NuminaMath

¹Detailed setup and pipelines (e.g., Q→Sol; Q→Sol1→Reflect/Refine→Sol2; multi-hypothesis with reflection) are described in our long-thought plan; evaluation uses GSM8K and MATH-500 (math-hard) with continuous pretraining (700B base + 50B tokens).

(Beeching et al., 2024) as our seed corpus and generate synthetic data for each step following the format in Figure 12.20:

- **Hypothesis (\mathcal{H}):** Based on the given question, generate a plan in natural language to solve the problem step-by-step. We start with single hypothesis generation and scale up to 3 hypothesis for each question.

You are an exceptional math teacher. Give a question, you teach the logic behind the solution to the question and the students utilize them to solve the problem. Given the Question above, write down three thought processes or hypotheses step-by-step in details that can be used parallelly to solve the question. DONOT SOLVE THE QUESTION. You MUST USE the latex notations properly for the equations. Your answer should follow the pattern below:
Hypothesis 1: <Detailed Step-by-Step Hypothesis>
Hypothesis 2: <Detailed Step-by-Step Hypothesis>
Hypothesis 3: <Detailed Step-by-Step Hypothesis>

- **Answer (\mathcal{A}):** Using the question and each hypothesis, LLM finds the answer by executing each step of the hypothesis sequentially. If there are multiple hypotheses, we generate multiple answers.

You are an exceptional math solver. Given the Question and Hypothesis above, provide solution using the Hypothesis by breaking down each equation and theorem step-by-step. Make sure that your solution strictly follows the Hypothesis. You MUST USE the latex notations properly for the equations.

- **Reflection (\mathcal{R}):** Evaluating each answer and the ground truth solution to see if any mistakes were made, any sub-goals were achieved, or if alternative approaches should be considered instead. This step provides a verdict whether the answer is correct or incorrect and if incorrect it points towards the step where the answer went wrong. We iterate this process for multiple solutions for a single problem.

You are an expert error detector. Given the Question, verify the correctness of the Test Solution using the Ground Truth Solution above. EXAMINE each step of the Test Solution and FIND out systematically if there is any incorrect logic/equation/calculation by comparing and evaluating with the Ground Truth Solution. Write all of your findings in details and end your evaluation with a comment "The Solution is verdict" where the verdict can be Correct or Incorrect. Use proper LaTeX notation for all equations.

- **Refine (\mathcal{R}'):** Using the knowledge gained from 'Reflection' stage, LLM refines the incorrect answer using the ground truth solution as reference—responding with a final answer and terminating the CoT.

Given the Question, we have a Test Solution and Correct Solution. If the Test Solution does not match with the Correct Solution, REFINE it step-by-step in details following the Correct Solution. Use proper LaTeX notation for all equations. ALWAYS end your answer with “Final Answer: The answer is $\boxed{\text{answer}}$ ”.

We tracked performance on GSM8K and math-hard benchmarks across these configurations. Results are summarized in [Table 12.26–Table 12.28](#). The ablation shows that adding explicit reflection and refine stages, and in multiple hypotheses, improves math-hard performance over question-to-solution-only baselines, indicating that **structured reasoning flow and logical sequence**—and the **conversational dynamics** of reflect-and-refine—are critical ingredients for complex math. When the best model from this setup is evaluated with R1-style few-shot prompts formatted as long thought ([Table 12.28](#)), the long-thought-pretrained model shows substantial gains on math-hard tasks compared to the baseline under the same prompt format. This supports that (i) the structure of the pretraining data (long thought with reflection/refine) transfers to inference-time structure (long-thought-style prompts), and (ii) for narrow, hard-math domains, **structure is as important as context**: the same prompt format that aligns with the pretraining structure unlocks the model’s capability.

Are the six properties ranked? The six attributes in this section are **not orthogonal**; they interact and their importance is **task-dependent**. For a *narrow domain* where the goal is to maximize performance on a specific task family (e.g., complex math, MATH-500), the following tend to be **core ingredients**: **Structured Reasoning Flow and Logical Sequence** (explicit steps, derivations, or multi-step pipelines) and **Conversational Dynamics as “Inner Monologue”** (reflection, refinement, role-based or iterative dialogue). These are the factors that distinguish MIND from rephrasing and that the long-thought ablation shows to matter for math-hard benchmarks. For *broader and scalable* gains—across domains, tasks, and with continuous generation of high-quality data—the following become **crucial**: **Cognitive Diversity and Multiple Solution Paths** (varied decomposition styles, multiple valid strategies), **Generalization Power and Domain Transfer** (benefits extending beyond the construction domain), and **Self-Expanding Potential** (synthetic frameworks that generate new structured examples from seeds). MIND already exhibits self-expanding potential (e.g., multiple conversational styles from one document) and generalization (gains on GSM8K, MATH, and general reasoning; [chapter 5](#)); CrossThink and related work demonstrate that cognitive diversity and domain transfer are essential for scaling reasoning beyond math ([chapter 6](#)). **Efficiency and Scalable Learning** (information density, Chinchilla-aware scaling) supports both narrow and broad goals by making the best use of token budget.

In summary, structure (reasoning flow, conversational dynamics, reflection–refine) is as important as context for complex reasoning; ablations over long-thought data show that reasoning components directly impact GSM8K and math-hard performance, and R1-style long-thought prompts amplify gains for models pretrained on long-thought data. There is no single ranking of the six properties; for narrow peak performance, structured flow and conversational dynamics are core, while for broad and scalable improvement, cognitive diversity, generalization, and self-expanding potential are crucial.

Configuration	Tokens (B)	GSM8k	MATH-500
NuminaMath (raw)	0.43	66.03	8.30
QA→Refine	1.60	63.15	9.65
Q→1 Hyp→Sol→Refine	1.48	63.46	8.87
Q→3 Hyp→3 Sol	2.40	64.52	8.68
Q→3 Hyp→3 Sol→3 Refine	4.10	62.40	8.52
Q→3 Hyp→3 Sol→3 Refine→GT	5.15	61.71	8.18

Table 12.26: **Long-thought ablation (without tags)**: Impact of reasoning components on GSM8K and MATH-500 (math-hard). Configurations vary from question-to-solution only to multi-hypothesis with reflect and refine.

Configuration	Tokens (B)	GSM8k	MATH-500
NuminaMath (raw)	0.43	65.20	8.93
QA→Refine	1.60	63.99	10.89
Q→1 Hyp→Sol→Refine	1.48	64.82	7.88
Q→3 Hyp→3 Sol→GT	2.40	64.97	10.11
Q→3 Hyp→3 Sol→3 Refine	4.10	65.81	11.04
Q→3 Hyp→3 Sol→3 Refine→GT	5.15	65.50	10.13

Table 12.27: **Long-thought ablation (with tags)**: Having multiple hypotheses, solutions and refinement phases in each solution yields the best math performances.

Model	GSM8k	Math-hard	Math-hard (R1)	MATH-500 (R1)
Base	79.45	19.48	49.92	69.60
NuminaMath (raw)	78.70	20.10	49.78	65.80
Q→3 Hyp→3 Sol→3 Ref	78.77	20.48	54.44	70.80

Table 12.28: **R1-style few-shot comparison**: When few-shot prompts are formatted as long thought (R1-style), the long-thought-pretrained model (NuminaMath-ours: Q→3 Hyp→3 Sol→3 Ref) shows gains on math-hard and MATH-500 (R1 Style) over the baseline and over NuminaMath-only pretraining.

Chapter 13

Appendix for Effect in Data Integration

This appendix supports [Part II](#), “Effect of reasoning data across LLM development phases,” and in particular [chapter 7](#) on the synergy between pretraining and post-training data. It includes a **breakdown of pretraining results** across individual benchmarks (ARC-C, RACE, GSM8K, Math-500, MMLU, HumanEval, and related tasks) for base and reasoning-augmented models ($\mathcal{M}_{\text{base}}$, \mathcal{M}_{SHQ} , \mathcal{M}_{LDQ} , \mathcal{M}_{LMQ}), showing that gains are broad and largest on math and code. Further sections cover **additional ablations** and extended experimental results that complement the main-chapter findings on front-loaded reasoning data, data diversity, and the asymmetric use of pretraining vs. fine-tuning.

13.1 Experiments and Results

Breakdown of Pretraining Results. [Table 13.1](#) provides a fine-grained view of the pretraining effects across individual benchmarks, complementing the domain-level averages reported in [Table 7.1](#). The improvements are not confined to a small subset of tasks but are broadly distributed, with the largest gains concentrated in mathematically intensive and program synthesis benchmarks. For example, on GSM8K and MATH-500, models trained with large-scale reasoning data exhibit dramatic jumps over $\mathcal{M}_{\text{base}}$, with \mathcal{M}_{LDQ} and \mathcal{M}_{LMQ} more than doubling performance on MATH-500. Similar trends appear in code generation tasks such as HUMAN-EVAL, HUMAN-EVALPLUS, and MBPP, where reasoning-augmented models consistently outperform the baseline, indicating stronger procedural and compositional capabilities.

In contrast, gains on general-purpose reasoning benchmarks such as ARC-C, RACE, WINOGRANDE, and HELLA-SWAG are more modest, suggesting that these tasks are less sensitive to explicit reasoning supervision during pretraining. Nevertheless, even in these cases, performance remains stable or slightly improved, indicating that incorporating reasoning data does not harm broad language understanding. Overall, the benchmark-level results reinforce the conclusion that early exposure to diverse reasoning corpora yields systematic improvements, with the most pronounced benefits emerging on tasks that require multi-step deduction, symbolic manipulation, or structured program synthesis.

Benchmark	$\mathcal{M}_{\text{base}}$	\mathcal{M}_{SHQ}	\mathcal{M}_{LDQ}	\mathcal{M}_{LMQ}	\mathcal{M}_{res}
ARC-C	80.89	80.46	81.40	81.83	81.15
RACE	73.59	75.41	78.28	79.43	76.68
WINOGRANDE	70.64	71.43	69.53	69.38	70.25
HELLASWAG	77.38	77.06	76.69	76.67	76.95
GSM8K	59.74	65.20	82.71	85.14	73.20
MATH-500	34.60	40.00	68.40	59.60	50.65
MMLU	61.67	61.45	65.87	65.42	63.60
MMLU-PRO	32.59	32.34	42.89	43.56	37.85
HUMANEVAL	37.44	41.04	48.63	51.68	44.70
HUMANEVALPLUS	32.59	35.03	42.74	46.28	39.16
MBPP	41.64	47.47	48.85	51.47	47.36
MBPP[SANITIZED]	51.87	53.74	59.53	60.97	56.53
MATH _{PT} AVG	47.17	52.60	75.56	72.37	61.92
SCIENCE _{PT} AVG	47.13	46.90	54.38	54.49	50.72
CODE _{PT} AVG	40.89	44.32	49.94	52.60	46.94
GPR _{PT} AVG	75.63	76.09	76.48	76.83	76.25
Overall	52.70	54.98	64.09	64.07	61.05

Table 13.1: Breakdown of base model accuracies across benchmarks. With increasing diversity and quality, the difference between $\mathcal{M}_{\text{base}}$ and models pretrained with reasoning data increases.

SFT Dataset: \mathcal{D}_{SHQ}					
Benchmark	$\mathcal{M}_{\text{base}} + \text{SFT}$	$\mathcal{M}_{\text{SHQ}} + \text{SFT}$	$\mathcal{M}_{\text{LDQ}} + \text{SFT}$	$\mathcal{M}_{\text{LMQ}} + \text{SFT}$	$\mathcal{M}_{\text{res}} + \text{SFT}$
IFEVAL	30.59	34.06	46.79	49.82	43.56
AIME-24	8.12	18.33	35.21	41.88	31.81
AIME-25	11.88	18.12	29.38	33.12	26.87
GSM8K	81.24	86.58	91.05	92.84	90.16
MATH-500	69.90	79.05	87.50	90.85	85.80
MMLU	52.14	62.90	71.15	73.49	69.18
MMLU-PRO	39.45	48.63	53.45	55.54	52.54
GPQA-DIAMOND	15.91	8.46	27.40	32.20	22.69
LIVECODEBENCH	10.48	24.76	28.57	35.55	29.63
MATH _{SFT} AVG	42.79	50.52	60.79	64.67	58.66
SCIENCE _{SFT} AVG	35.83	40.00	50.67	53.74	48.14
CODE _{SFT} AVG	10.48	24.76	28.57	35.55	29.63
INS _{SFT} AVG	30.59	34.06	46.79	49.82	43.56
Overall	35.52	42.32	52.28	56.14	50.25

Table 13.2: Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{SHQ} .

Detailed SFT Benchmark Performance. While the Table 7.2 presents aggregate performance metrics across different reasoning domains, Table 13.2, 13.3, and 13.4 provide a granular breakdown of how each reasoning model performs across individual benchmarks when fine-tuned on \mathcal{D}_{SHQ} , \mathcal{D}_{LDQ} , and \mathcal{D}_{LMQ} respectively.

SFT Dataset: \mathcal{D}_{LDQ}					
Benchmark	$\mathcal{M}_{\text{base}} + \text{SFT}$	$\mathcal{M}_{\text{SHQ}} + \text{SFT}$	$\mathcal{M}_{\text{LDQ}} + \text{SFT}$	$\mathcal{M}_{\text{LMQ}} + \text{SFT}$	$\mathcal{M}_{\text{res}} + \text{SFT}$
IFEVAL	50.86	47.01	57.61	59.21	54.61
AIME-24	1.15	2.50	6.37	4.90	4.59
AIME-25	0.83	3.12	7.71	9.38	6.74
GSM8K	73.56	75.11	59.81	77.62	70.84
MATH-500	46.70	44.98	39.63	56.28	46.96
MMLU	15.25	9.95	49.15	56.81	38.64
MMLU-PRO	16.26	14.24	30.50	33.51	26.08
GPQA-DIAMOND	8.97	7.39	26.01	28.35	20.58
LIVECODEBENCH	6.04	10.48	10.16	11.91	10.85
MATH _{SFT} AVG	30.56	31.43	28.38	37.04	32.28
SCIENCE _{SFT} AVG	13.49	10.52	35.22	39.55	28.43
CODE _{SFT} AVG	6.04	10.48	10.16	11.91	10.85
INS _{SFT} AVG	50.86	47.01	57.61	59.21	54.61
Overall	25.24	24.86	32.84	36.93	31.54

Table 13.3: Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{LDQ} .

SFT Dataset: \mathcal{D}_{LMQ}					
Benchmark	$\mathcal{M}_{\text{base}} + \text{SFT}$	$\mathcal{M}_{\text{SHQ}} + \text{SFT}$	$\mathcal{M}_{\text{LDQ}} + \text{SFT}$	$\mathcal{M}_{\text{LMQ}} + \text{SFT}$	$\mathcal{M}_{\text{res}} + \text{SFT}$
IFEVAL	50.50	52.65	57.78	58.79	56.41
AIME-24	1.25	3.13	8.23	4.69	5.35
AIME-25	0.84	2.92	6.98	7.09	5.66
GSM8K	72.93	74.03	57.70	76.27	69.33
MATH-500	45.33	42.18	36.93	50.75	43.28
MMLU	15.75	6.90	50.90	55.15	37.65
MMLU-PRO	15.57	13.18	32.09	33.37	26.21
GPQA-DIAMOND	8.97	4.87	23.17	29.99	19.34
LIVECODEBENCH	4.76	9.37	10.95	9.05	9.79
MATH _{SFT} AVG	30.09	30.56	27.46	34.70	30.91
SCIENCE _{SFT} AVG	13.43	8.31	35.39	39.50	27.73
CODE _{SFT} AVG	4.76	9.37	10.95	9.05	9.79
INS _{SFT} AVG	50.50	52.65	57.78	58.79	56.41
Overall	24.69	25.22	32.89	35.51	31.21

Table 13.4: Breakdown of model accuracies across benchmarks after training SFT phase on the \mathcal{D}_{LMQ} .

These detailed results confirm that the “reasoning advantage”—established during the 1T pretraining phase—is not limited to aggregate scores but persists across all distinct evaluation categories, including AIME, GSM8K, MMLU, and LIVECODEBENCH. Models pretrained with reasoning-specific corpora (\mathcal{M}_{LDQ} and \mathcal{M}_{LMQ}) consistently outperform $\mathcal{M}_{\text{base}}$ and \mathcal{M}_{SHQ} across these benchmarks, regardless of the SFT dataset used. Specifically, $\mathcal{M}_{LMQ} + \text{SFT}$ frequently achieves the highest peak performance in complex reasoning tasks like AIME-24/25 and MATH-500, suggesting that high-quality, dense reasoning pretraining provides a more robust initialization that SFT can more effectively refine.

Model	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{SHQ}}$	42.79	35.83	10.48	30.59
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{LDQ}}$	30.56	13.49	6.04	50.86
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{LMQ}}$	30.09	13.43	4.76	50.50
$\mathcal{M}_{\text{base}} + \text{SFT}$	34.48	20.92	7.09	43.98
$\mathcal{M}_{\text{SHQ}} + \text{SFT}_{\text{SHQ}}$	50.52	40.00	24.76	34.06
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{SHQ}}$	60.79	50.67	28.57	46.79
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{SHQ}}$	64.67	53.74	35.55	49.82
$\mathcal{M}_{\text{SHQ}} + \text{SFT}_{\text{LDQ}}$	31.43	10.52	10.48	47.01
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{LDQ}}$	28.38	35.22	10.16	57.61
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{LDQ}}$	37.04	39.55	11.91	59.21
$\mathcal{M}_{\text{SHQ}} + \text{SFT}_{\text{LMQ}}$	30.56	8.31	9.37	52.65
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{LMQ}}$	27.46	35.39	10.95	57.78
$\mathcal{M}_{\text{LMQ}} + \text{SFT}_{\text{LMQ}}$	34.70	39.50	9.05	58.79
$\mathcal{M}_{\text{res}} + \text{SFT}$	40.62	34.77	16.75	51.52

Table 13.5: **Results of all SFT models with varying pretraining and SFT data.** Model pretrained with reasoning data obtains the highest gain after SFT phase of training.

Model	MATH _{PT} AVG	SCIENCE _{PT} AVG	CODE _{PT} AVG	GPR _{PT} AVG	Overall Avg
$\mathcal{M}_{\text{base}}$	6.92	18.91	12.82	40.92	19.89
\mathcal{M}_{LMQ}	16.39	21.50	23.44	41.69	25.75

Table 13.6: **Effect of early reasoning exposure on a 1.2B Transformer model.** The reasoning-augmented model improves substantially on math, science, and code benchmarks while maintaining general-domain performance.

To facilitate a direct comparison of how different SFT strategies interact with varying pretraining foundations, Table 13.5 aggregates the results across all model-SFT combinations. The data underscores a clear synergy: the best overall performance is achieved when models are pretrained on diverse, high-quality reasoning data and subsequently fine-tuned on a compatible SFT dataset. This highlights that while SFT is crucial for aligning model behavior, the “foundation” built during pretraining serves as the primary determinant for the model’s upper bound on complex reasoning capabilities. Even when given the same SFT budget, models lacking reasoning-specific pretraining struggle to close the performance gap, particularly in challenging domains like code generation and advanced mathematics.

13.2 Additional Ablations

Generalization and Robustness Across Model Scales and Architectures. To ensure transparency and reproducibility, all pretraining, SFT, and RL datasets—including Common Crawl, Arxiv, Wikipedia, StackExchange, GitHub, OpenWebText, and OpenWebMath—are derived from fully open corpora. To validate the external validity of our data strategy and the effects of front-loading reasoning data across varying scales and architectures, we extended our experiments to a Transformer-based model \mathcal{M} with 1.2B parameters trained on 125B tokens. We compared the baseline $\mathcal{M}_{\text{base}}$ against \mathcal{M}_{LMQ} , which integrates early reasoning exposure from \mathcal{D}_{LMQ} while maintaining a constant token ratio between $\mathcal{D}_{\text{base}}$ and \mathcal{D}_{LMQ} .

As summarized in Table 13.6, the integration of reasoning data consistently enhances task-specific performance across different architectures. We observed substantial gains, specifically +9.47% in Math, +2.60% in Science, and +10.62% in Code benchmarks. Critically, these enhancements do not degrade generalized capabilities; performance on general benchmarks (GPR_{PT} AVG) remains stable, indicating that early exposure to reasoning data does not perturb the model’s fundamental language modeling objectives. These findings provide strong empirical evidence that our data strategy is robust and scalable, demonstrating that front-loading reasoning data is an effective mechanism for augmenting complex deduction capabilities regardless of the underlying model scale or architecture.

Anatomy of high-quality reasoning data in SFT. Our previous results establish that SFT benefits immensely from high-quality data, but what precisely constitutes “quality” remains unclear. In this ablation, we investigate a defining characteristic of such data: the depth and complexity of its reasoning traces. Specifically, we compare datasets that differ both in reasoning length and construction method. The high-quality corpus \mathcal{D}_{SHQ} consists of answers generated by strong teacher models, characterized by long chain-of-thoughts with an average length exceeding 10k tokens. In contrast, \mathcal{D}_{LDQ} provides reasoning data from diverse domains but with much shorter and noisier reasoning traces (average \sim 550 tokens). This distinction highlights a potential mechanism underlying quality: longer reasoning chains may serve as richer supervisory signals, encouraging models to internalize structured multi-step inference rather than surface-level heuristics.

To test this hypothesis, we extract from \mathcal{D}_{LLQ} only the longest reasoning traces, creating a new dataset \mathcal{D}_{ALF} . Although it represents only \sim 2% of the original \mathcal{D}_{LLQ} corpus, \mathcal{D}_{ALF} is highly skewed toward domains with inherently deeper reasoning (75% math, with the remainder in science, code, and general reasoning). We then conduct SFT on top of the \mathcal{M}_{llq} model using both \mathcal{D}_{LLQ} (quantity and diversity) and \mathcal{D}_{ALF} (length-filtered complexity).

Model	Average	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{LDQ}}$	32.84	28.38	35.22	10.16	57.61
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{ALF}}$	42.71	60.95	47.50	22.54	39.87

Table 13.7: **Impact of depth in reasoning traces in data on SFT phase.** Model trained on longer CoT reasoning data outperforms the one trained on diverse reasoning traces.

As shown in Table 13.7, emphasizing depth in reasoning traces has a significant impact on downstream reasoning tasks. While finetuning with \mathcal{D}_{LLQ} yields only modest improvements, switching to the 50 times smaller, filtered by reasoning depth via answer length \mathcal{D}_{ALF} boosts the overall score to 9.87%, with particularly strong gains in math, science and code. Interestingly, this comes at the cost of slightly reduced accuracy on instruction-following tasks, reflecting a trade-off between breadth and reasoning-specific depth. These results provide strong evidence that *longer chain-of-thought supervision is a critical marker of quality in SFT data*. Even when drawn from a noisy, large-scale corpus, selecting for reasoning depth alone can yield outsized improvements, making length-filtering a simple yet cost-effective heuristic for constructing impactful reasoning datasets for SFT phase.

Data Redundancy Reinforces Foundational Skills, Not Overfitting. A critical consideration in our two-phase approach is whether using the same reasoning data in both pretraining and SFT leads to catastrophic forgetting or brittle overfitting, a known concern in sequential fine-tuning (Luo et al., 2025b; Chen et al., 2025a).

Our results, shown in Figure 13.1, suggest this concern is unfounded and that the opposite is true: for reasoning, strategic redundancy is highly beneficial. The baseline model, \mathcal{M}_{base} , exposed to the high-quality \mathcal{D}_{SHQ} data only during SFT, is the lowest performer across all categories. In contrast, \mathcal{M}_{SHQ} , which sees this same data in both phases, demonstrates a significant performance uplift, indicating that the second exposure reinforces rather than overwrites learning. We hypothesize this occurs because the two training phases serve different learning functions. During pretraining, the reasoning data is integrated slowly into the model’s core representations alongside vast, diverse knowledge, forcing an internalization of abstract logical patterns.

The SFT phase then acts not as a new learning task, but as a powerful reinforcement signal on an already-prepared foundation. This benefit is amplified by a diverse pretraining context: the top-performing \mathcal{M}_{LMQ} model leverages its broad exposure to various reasoning styles to most effectively capitalize on the repeated, high-quality signal from \mathcal{D}_{SHQ} . This suggests that data redundancy between pretraining and SFT should be viewed as a powerful mechanism for skill consolidation, where a diverse pretraining builds the capacity for reasoning and redundant SFT sharpens it.

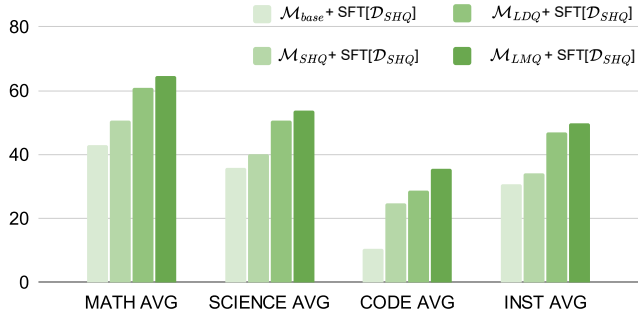


Figure 13.1: The model that saw the same high-quality data in both pretraining and SFT (\mathcal{M}_{SHQ}) handily beats the baseline (\mathcal{M}_{base}) that only saw the data once.

Pretrained Foundations vs. Scaling SFT Data. In Table 7.4, the notation $\text{SFT} \times 2$ refers to extending the SFT phase for twice the number of epochs over the existing dataset, rather than expanding the training set with unique samples. To further investigate whether reasoning performance is an artifact of data quantity—and to address potential concerns regarding overfitting with repeated data—we conducted a controlled experiment comparing a no-reason base model against our reason-aware baseline. We examined whether the no-reason base model, \mathcal{M}_{base} , could recover the performance of the reason-aware model, \mathcal{M}_{LDQ} , by utilizing a significantly larger volume of unique SFT tokens. We fine-tuned \mathcal{M}_{base} on a combined dataset of \mathcal{D}_{LDQ} and \mathcal{D}_{ALF} (totaling 268M unique samples) and compared it against \mathcal{M}_{LDQ} , which was fine-tuned solely on \mathcal{D}_{ALF} (7.1M unique samples). Both models were evaluated under the identical SFT evaluation setup.

As demonstrated in Table 13.8, the reason-aware base model achieves a 39% relative improvement in the Overall score and consistently outperforms the no-reason baseline across all domains, despite the latter receiving substantially more unique training data. This result highlights that strong reasoning foundations established during pretraining cannot

Model	MATH _{SFT} AVG	SCIENCE _{SFT} AVG	CODE _{SFT} AVG	INS _{SFT} AVG	Overall
$\mathcal{M}_{\text{base}} + \text{SFT}_{\text{LDQ}+\text{ALF}}$	33.66	29.15	3.49	56.86	30.79
$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{ALF}}$	60.95	47.29	22.54	39.87	42.66

Table 13.8: **Comparison between SFT with large amounts of unique data and reasoning-augmented pretraining.** The reasoning-based model substantially outperforms the no-reason baseline despite using significantly fewer unique SFT samples.

Instruction Type	$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{ALF}}$	$\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{LDQ}}$	Diff
punctuation:no_comma	13	27	14
length_constraints:number_words	19	30	11
change_case:english_lowercase	19	28	9
keywords:letter_frequency	13	21	8
change_case:english_capital	9	16	7
language:response_language	19	26	7
detectable_format:number_bullet_lists	14	21	7
combination:two_responses	11	16	5
keywords:forbidden_words	12	17	5
detectable_format:title	32	36	4
startend:quotation	19	23	4
change_case:capital_word_frequency	12	15	3
length_constraints:number_paragraphs	9	12	3
length_constraints:nth_paragraph_first_word	3	5	2
length_constraints:number_sentences	29	31	2
detectable_format:json_format	13	15	2
startend:end_checker	11	13	2
detectable_format:number_highlighted_sections	40	41	1
detectable_format:constrained_response	9	10	1
detectable_content:number_placeholders	23	23	0
combination:repeat_prompt	17	17	0
detectable_format:multiple_sections	12	12	0
detectable_content:postscript	23	22	-1
keywords:existence	31	30	-1
keywords:frequency	29	26	-3

Table 13.9: Instruction-wise accuracy comparison on IFEval between models trained with less diverse (\mathcal{D}_{ALF}) and more diverse (\mathcal{D}_{LDQ}) corpora.

be trivially recovered through additional SFT, even when the model is provided with a significantly larger scale of unique data.

Breadth vs. Alignment: The Role of Data Diversity in Instruction Following. [Table 7.5](#)

reveals a consistent trade-off: length-filtered long-CoT SFT on \mathcal{D}_{ALF} improves reasoning performance while weakening instruction-following ability. This pattern suggests that reduced data diversity affects alignment-sensitive behaviors. We hypothesize that this degradation stems from distributional skew and stylistic rigidity, as \mathcal{D}_{ALF} is heavily concentrated in math and code domains, similar in structure to \mathcal{D}_{SHQ} . While such data reinforces precise token-level reasoning, it exposes the model to a narrower range of linguistic forms and formatting variations. To investigate this phenomenon, we compare instruction-level accuracy on the IFEval benchmark between two models: $\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{ALF}}$ (less diverse SFT) and $\mathcal{M}_{\text{LDQ}} + \text{SFT}_{\text{LDQ}}$ (diverse SFT).

Model	Overall Avg	MATH _{PT} AVG	SCIENCE _{PT} AVG	CODE _{PT} AVG	GPR _{PT} AVG
\mathcal{M}_{LDQ}	64.09	75.56	54.38	49.94	76.48
\mathcal{M}_{ALF}	52.59	46.73	46.03	41.66	75.96

Table 13.10: **Effect of reducing scale and diversity of reasoning data during pretraining.** \mathcal{M}_{ALF} is trained on a length-filtered subset of \mathcal{D}_{LDQ} that is smaller and less diverse.

As shown in Table 13.9, the diverse \mathcal{D}_{LDQ} dataset yields substantial improvements in linguistic manipulation tasks, including punctuation control, case transformation, word-count constraints, and formatting instructions. These categories require flexibility in natural language generation, sensitivity to stylistic variation, and the ability to adapt output structure. Exposure to diverse reasoning data introduces broader linguistic cues, Markdown structures, and formatting styles, enabling stronger control over output form.

In contrast, the math- and code-heavy \mathcal{D}_{ALF} dataset reinforces strict token precision and pattern adherence. This benefits tasks involving keyword presence, token counting, or rigid structural constraints but provides limited stylistic variability. As a result, models trained on less diverse data excel at “hard” lexical constraints while underperforming on “soft” linguistic transformations that require adaptability. These findings clarify that data diversity plays a critical role in preserving instruction-following capabilities. Reasoning gains from length-filtered long-CoT data come with a narrowing of stylistic exposure, and broader linguistic coverage helps maintain alignment-related flexibility without sacrificing reasoning performance.

Controlling for Scale and Diversity in Reasoning Pretraining. To isolate the role of scale and diversity from other potential confounders such as dataset source or content differences, we conduct a controlled pretraining experiment using a downsampled subset of \mathcal{D}_{LDQ} . Specifically, we construct \mathcal{D}_{ALF} by retaining only examples whose answer length exceeds 4096 tokens. This filtering procedure selects long-chain-of-thought samples that are typically math- and code-centric, resulting in data that is arguably high quality in reasoning depth but substantially smaller in scale and less diverse in topical coverage than the full \mathcal{D}_{LDQ} corpus. During pretraining of \mathcal{M}_{ALF} , we maintain the same token ratio between \mathcal{D}_{base} and \mathcal{D}_{ALF} as in the original setup to ensure a controlled comparison.

As detailed in Table 13.10, the model trained on the downsampled subset exhibits an absolute 11.5 % drop in overall pretraining accuracy compared to \mathcal{M}_{LDQ} . The degradation is particularly pronounced in math and code benchmarks, while general-domain performance remains relatively stable. These results indicate that although long-chain-of-thought samples capture high-quality reasoning traces, reducing dataset scale and diversity substantially weakens the overall pretraining signal.

This controlled comparison supports the claim that reasoning quality alone is insufficient to explain performance gains. Instead, diversity and scale play a dominant role during pretraining, enabling broader generalization across reasoning-intensive domains.

Chapter 14

Appendix for Redesigning Pipeline

This appendix accompanies [Part III](#), “Redesign the Learning Pipeline to Enable Reasoning Emergence,” and supplies details for both [chapter 9](#) (Difference-Masking) and [chapter 10](#) (Reinforcement as Pretraining Objective). **Difference-Masking**: detailed experimental settings (hyperparameters for continuous pretraining and fine-tuning, language and multimodal); masking of video tokens (bounding-box strategies, TVQA and Social-IQ results); and masking of language tokens. **RLP**: formal **proofs** for the tokenwise cross-entropy reduction identity, the lower bound via marginalization over thoughts, and the tokenwise-to-sequence connection under teacher forcing; analysis of why relative advantages do not reward bad thoughts (with monotonic-improvement proof and numerical illustration); experimental setup; extended ablation details; additional ablations; and data-blend extended results.

14.1 Detailed Experimental Settings

In this section, we provide an overview of the experimental conditions utilized in our study. To ensure fair comparisons with our baselines, we maintain a consistent set of hyperparameters for both continuous pretraining and fine-tuning. For language tasks, we largely adhere to the hyperparameters employed in [Gururangan et al. \(2020\)](#). Throughout our experiments, we maintain a masking ratio of 25% in both language and multimodal settings. We adopt a static masking strategy, replacing masked tokens with random values.

Hyperparameters	CPT		FT	
	Language	Multimodal	Language	Multimodal
<i>learning_rate</i>	0.0001	0.000005	1.00E-06	5.00E-06
<i>num_train_epochs</i>	150	20	10	20
<i>eval_every_n_epochs</i>	30	1	1	1
<i>patience</i>	20	5	3	5

Table 14.1: List of hyperparameters used in both continuous pretraining (CPT) and finetuning (FT).

We reproduce MERLOT-Reserve’s original training on TVQA: we decompose samples in Social-IQ and TVQA from the form (Question, All Answers, Video Information) into a list of 3-tuples: (Question, Candidate Answer, Video Information). MERLOT scores each candidate answer independently, given the question and video, and is trained with loss that

encourages the model to minimize estimated likelihood of incorrect answers and maximize likelihood of correct answers.

From video frames, we mask image patches into 16x16 patches as determined by MERLOT-Reserve’s backbone image transformer ViT (Dosovitskiy et al., 2021). The language experiments took nine hours of runtime each on a single 12GB GPU, and the multimodal vision experiments required six hours on a single TPU v2-8.

14.2 Masking Video Tokens

Following the intuition from language, we hypothesize that masking and predicting small patches of an image may be testing *local* capabilities (e.g. determining what an eye looks like from the rest of the face) rather than *global* capabilities (e.g. determining what a person’s face looks like from the rest of the scene, including other people’s faces).

Accordingly, instead of masking low-level image patches, we mask groups of patches corresponding to a higher level semantic entity: bounding boxes over objects in the image. We see this approach as a visual analogue for masking at the word-level instead of the token-level in our language experiments. We found that $K = 1$ performed much better than other values, where the selected anchor word was “person”. We considered two possible bounding boxes associated with people: bounding boxes over faces and bodies. We evaluated both options and found that considering entire bounding boxes over people’s bodies (including their faces) performed the best. These results are shown in Table 14.2.

Masking Strategy	TVQA	Social-IQ
Random Masking	73.75	69.05
DIFFERENCE-MASKING (Face)	81.51	69.13
DIFFERENCE-MASKING (Body)	81.73	71.37

Table 14.2: Results of DIFFERENCE-MASKING on multimodal video understanding benchmarks TVQA and Social IQ. DIFFERENCE-MASKING leads to an improvement of 8% and 2% accuracy over random accuracy.

We extracted body detection coordinates using UniTrack (Wang et al., 2021) and face detection coordinates using MTCNN (Zhang et al., 2016a).

Apart from the bounding box strategy, we also experimented with masking patches chosen by differences between CLIP embeddings Radford et al. (2021b) of the anchor and the vision patch directly (without bounding box labels). Our experiments validate that the CLIP-based masking strategy performs poorly compared to our bounding box strategy. One possible reason can be that CLIP is not robust enough for video datasets which led to masking patches that are not relevant to the anchor word “person”.

14.3 Masking Language Tokens

In Section 9.3.3 we describe the motivation for using a word-level strategy in our implementation of DIFFERENCE-MASKING. An alternative implementation could be to assign each token in a word the same masking likelihood, and mask tokens only by this probability.

	TVQA	Social-IQ
CLIP Radford et al. (2021b)	73.58	68.75
DIFFERENCE-MASKING	81.73	71.37

Table 14.3: We validate our hypothesis that masking patches using DIFFERENCE-MASKING is more effective than masking using CLIP similarity.

This could result in some tokens from the same word being masked where others are not. Our intuition is that for specialized domains such as chemistry, subword tokens may be trivial to predict from their neighbors, but whole words may not be trivial to predict given the context. For example, a word such as “phosphates” would be tokenized into “phos” and “-phates”. We expect that it may be trivial to predict “phos” given “-phates” or vice versa, but it may be hard (and may promote a better understanding of the task) to predict the word “phosphates” given the context.

Empirically, we find that this decision improved performance substantially, as shown in the results in Table 14.4 below.

	ACL-ARC	ChemProt
Token	0.6501	0.8224
Word	0.7404	0.8394

Table 14.4: We validate our hypothesis that masking tokens using DIFFERENCE-MASKING at the word-level is more effective than masking at the token-level.

14.3.1 Proofs

In this section, we provide the proofs supporting the methodology in §10.2. We first prove the tokenwise cross-entropy (CE) reduction identity (Prop. 1), then the lower bound via marginalization over thoughts (Prop. 2). Finally, we state and prove Prop. 3, which formalizes the positionwise-credit claim described in §10.2.5: under teacher forcing, averaging the expected tokenwise information-gain rewards across positions recovers the expected per-token sequence-level CE improvement.

For convenience, we recall the key definitions from the main text: the reasoned and baseline log-evidence $S_{\text{pred}}(c_t) = \log p_{\theta}(x_t | x_{<t}, c_t)$ and $S_{\text{EMA}} = \log \bar{p}_{\phi}(x_t | x_{<t})$ (equation 10.3); the information-gain reward $r(c_t) = S_{\text{pred}}(c_t) - S_{\text{EMA}}$ (equation 10.5); and the cross-entropy $\text{CE}(p, q) \stackrel{\text{def}}{=} \mathbb{E}_{x \sim p}[-\log q(x)]$ (equation 10.2).

14.3.2 Proof of Proposition 1 (Expected improvement identity)

Proof of Proposition 1. Fix the context $x_{<t}$ and a realized thought c_t , and let $p_t^*(x) := p^*(x | x_{<t})$ denote the data distribution over x_t at this position. By the reward definition equation 10.5 together with equation 10.3,

$$r(c_t) = \log p_{\theta}(x_t | x_{<t}, c_t) - \log \bar{p}_{\phi}(x_t | x_{<t}).$$

Taking expectation with respect to $x_t \sim p_t^*$ and using linearity of expectation,

$$\mathbb{E}_{x_t \sim p_t^*} [r(c_t)] = \mathbb{E}_{x_t \sim p_t^*} [\log p_\theta(x_t | x_{<t}, c_t)] - \mathbb{E}_{x_t \sim p_t^*} [\log \bar{p}_\phi(x_t | x_{<t})].$$

By the definition of cross-entropy equation 10.2, $\text{CE}(p, q) = \mathbb{E}_{x \sim p}[-\log q(x)]$, so each expectation of a log-likelihood equals the negative cross-entropy:

$$\begin{aligned} \mathbb{E}_{x_t \sim p_t^*} [\log p_\theta(x_t | x_{<t}, c_t)] &= -\text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)), \\ \mathbb{E}_{x_t \sim p_t^*} [\log \bar{p}_\phi(x_t | x_{<t})] &= -\text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})). \end{aligned}$$

Substituting into the previous display yields

$$\mathbb{E}_{x_t \sim p_t^*} [r(c_t)] = \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})) - \text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)),$$

which is the desired identity. \square

14.3.3 Proof of Proposition 2 (Lower bound via marginalization over thoughts)

Proof of Proposition 2. Fix $(x_{<t}, x_t)$ and recall $S_{\text{pred}}(c_t) = \log p_\theta(x_t | x_{<t}, c_t)$ and $\tilde{p}_\theta(x | x_{<t}) = \mathbb{E}_{z_t \sim \pi_\theta(\cdot | x_{<t})} [p_\theta(x | x_{<t}, z_t)]$.

(i) Jensen bound. Conditioning on $(x_{<t}, x_t)$ and taking expectation over $c_t \sim \pi_\theta(\cdot | x_{<t})$, $\mathbb{E}_{c_t \sim \pi_\theta} [S_{\text{pred}}(c_t)] = \mathbb{E}_{c_t} [\log p_\theta(x_t | x_{<t}, c_t)] \leq \log \mathbb{E}_{c_t} [p_\theta(x_t | x_{<t}, c_t)] = \log \tilde{p}_\theta(x_t | x_{<t})$, where the inequality is Jensen's inequality applied to the concave function $\log(\cdot)$. This proves (i) pointwise for the realized x_t .

(ii) Bound on $J(\theta)$. By definition of the reward in equation 10.5 and teacher forcing (see equation 10.3),

$$\begin{aligned} J(\theta) &= \mathbb{E} [\mathbb{E}_{c_t \sim \pi_\theta} [S_{\text{pred}}(c_t)] - S_{\text{EMA}}] \\ &\leq \mathbb{E} [\log \tilde{p}_\theta(x_t | x_{<t}) - \log \bar{p}_\phi(x_t | x_{<t})] \\ &= \mathbb{E} [\log \tilde{p}_\theta(x_t | x_{<t}) - \log \bar{p}_\phi(x_t | x_{<t})], \end{aligned}$$

where the inequality uses part (i) and the outer expectation is over $(x_{<t}, x_t) \sim \mathcal{D}$. This proves (ii).

Tightness. Equality in (i) (and hence in (ii)) holds precisely when $p_\theta(x_t | x_{<t}, c_t)$ is almost surely constant in c_t under $\pi_\theta(\cdot | x_{<t})$ (e.g., when the predictor ignores the thought or when the thought policy is degenerate). \square

14.3.4 Tokenwise-to-sequence connection under teacher forcing (positionwise credit)

This subsection formalizes the claim in §10.2.5 that summing positionwise CE improvements recovers the sequence-level (per-token) improvement. The following proposition is *new to the appendix* and not required elsewhere; it clarifies how tokenwise rewards aggregate at the sequence level under teacher forcing.

Proposition 3 (Tokenwise-to-sequence connection under teacher forcing). *Let $x = (x_1, \dots, x_T)$ be drawn from the data distribution $p^*(x)$ and fix a policy $\pi_\theta(c_t | x_{<t})$, the reasoned scorer $p_\theta(\cdot | x_{<t}, c_t)$, and the no-think baseline $\bar{p}_\phi(\cdot | x_{<t})$. Define the sequence-level (per-token) cross-entropy for the baseline and the (stochastic) reasoned scorer by*

$$\begin{aligned} \text{CE}_{\text{seq}}(p^*, \bar{p}_\phi) &:= \mathbb{E}_{x \sim \mathcal{D}} \left[-\frac{1}{T} \sum_{t=1}^T \log \bar{p}_\phi(x_t | x_{<t}) \right], \\ \text{CE}_{\text{seq}}(p^*, p_\theta[\pi_\theta]) &:= \mathbb{E}_{x \sim \mathcal{D}} \left[-\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{c_t \sim \pi_\theta(\cdot | x_{<t})} [\log p_\theta(x_t | x_{<t}, c_t)] \right]. \end{aligned}$$

Then the average over positions of the expected tokenwise information-gain rewards equals the per-token sequence-level CE improvement of the reasoned scorer against the baseline:

$$\mathbb{E}_x \left[\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{c_t \sim \pi_\theta(\cdot | x_{<t})} \mathbb{E}_{x_t \sim p^*(\cdot | x_{<t})} [r(c_t)] \right] = \text{CE}_{\text{seq}}(p^*, \bar{p}_\phi) - \text{CE}_{\text{seq}}(p^*, p_\theta[\pi_\theta]).$$

Proof. (i) **Conditional independence under teacher forcing.** At position t , teacher forcing samples the target token from the data channel while the thought is sampled from the policy given the same prefix:

$$x_t \sim p^*(\cdot | x_{<t}), \quad c_t \sim \pi_\theta(\cdot | x_{<t}).$$

Hence

$$p(c_t, x_t | x_{<t}) = \pi_\theta(c_t | x_{<t}) p^*(x_t | x_{<t}), \quad \text{i.e. } c_t \perp x_t | x_{<t}.$$

This implies $\mathbb{E}_{x_t \sim p^*(\cdot | x_{<t}, c_t)}[\cdot] = \mathbb{E}_{x_t \sim p^*(\cdot | x_{<t})}[\cdot]$.

(ii) **Positionwise CE reduction.** By Proposition 1, for any fixed $(x_{<t}, c_t)$,

$$\mathbb{E}_{x_t \sim p^*(\cdot | x_{<t})} [r(c_t)] = \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})) - \text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)).$$

Taking expectation over $c_t \sim \pi_\theta(\cdot | x_{<t})$ and using linearity of expectation gives

$$\mathbb{E}_{c_t} \mathbb{E}_{x_t} [r(c_t)] = \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})) - \mathbb{E}_{c_t} \text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)).$$

(iii) **Sum over positions.** Average the identity in (ii) over $t = 1, \dots, T$ and over $x \sim \mathcal{D}$:

$$\begin{aligned} &\mathbb{E}_x \left[\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{c_t} \mathbb{E}_{x_t} [r(c_t)] \right] \\ &= \mathbb{E}_x \left[\frac{1}{T} \sum_{t=1}^T \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})) \right] - \mathbb{E}_x \left[\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{c_t} \text{CE}(p^*, p_\theta(\cdot | x_{<t}, c_t)) \right]. \end{aligned}$$

By the definition of cross-entropy in equation 10.2 and the chain rule for likelihoods,

$$\mathbb{E}_{x_t \sim p^*(\cdot | x_{<t})} [-\log \bar{p}_\phi(x_t | x_{<t})] = \text{CE}(p^*, \bar{p}_\phi(\cdot | x_{<t})),$$

and similarly for the reasoned scorer inside the c_t -expectation. Therefore the two sums on the right are exactly $\text{CE}_{\text{seq}}(p^*, \bar{p}_\phi)$ and $\text{CE}_{\text{seq}}(p^*, p_\theta[\pi_\theta])$ as defined above, yielding the claimed equality. \square

14.4 Why Relative Advantages Do Not Reward Bad Thoughts

14.4.1 Proof of Monotonic Improvement

It may seem paradoxical that, when all thoughts perform poorly ($r(c_t) < 0$), the group-relative formulation still labels one as “better” and reinforces it. Does this mean the model is being trained to favor bad reasoning? We demonstrate that, mathematically, this mechanism is sound: the update remains an unbiased gradient step on $J(\theta)$, ensuring monotonic improvement even in such cases.

1.Objective. For context $x_{<t}$ and target token x_t :

$$J(\theta) = \mathbb{E}_{c \sim \pi_\theta} [r(c)], \quad r(c) = \log p_\theta(x_t | x_{<t}, c) - \log \bar{p}_\phi(x_t | x_{<t}). \quad (14.1)$$

Maximizing J reduces cross-entropy versus the no-think baseline. Ignoring stop-gradients, the policy gradient is

$$\nabla_\theta J(\theta) = \mathbb{E}_{c \sim \pi_\theta} [r(c) \nabla_\theta \log \pi_\theta(c)]. \quad (14.2)$$

2. Group-relative advantages are unbiased. We draw $G \geq 2$ thoughts $c^{(1)}, \dots, c^{(G)} \sim \pi_\theta$ and form

$$\bar{r} = \frac{1}{G} \sum_{j=1}^G r(c^{(j)}), \quad (14.3)$$

$$A^{(i)} = \frac{G}{G-1} (r(c^{(i)}) - \bar{r}). \quad (14.4)$$

Let $\mu = \mathbb{E}[r(c)]$. Then

$$\mathbb{E}[A^{(i)} | c^{(i)}] = r(c^{(i)}) - \mu, \quad (14.5)$$

and

$$\mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G A^{(i)} \nabla_\theta \log \pi_\theta(c^{(i)}) \right] = \nabla_\theta J(\theta). \quad (14.6)$$

Hence, the estimator is **unbiased**. Even if all rewards are negative, the update follows the correct gradient direction.

3. Why positive advantage for the “least-bad” rollout is correct. As the model learns, it gradually increases the probability of generating thoughts that help prediction and decreases the probability of those that do not. This process, known as the *replicator dynamic*, captures how relative advantages drive steady improvement over time:

$$\dot{\pi}(c) = \pi(c) [r(c) - \mu], \quad (14.7)$$

whose improvement rate is

$$\frac{d}{d\tau} J(\theta(\tau)) = \text{Var}_\pi[r(c)] \geq 0. \quad (14.8)$$

Even if all $r(c) < 0$, shifting probability mass from more-negative to less-negative thoughts *increases* J . Thus, a positive advantage for the least-bad thought reflects correct relative improvement, not misaligned reward.

4. Monotonic expected improvement. With unbiased gradient estimator $\hat{g} \approx \nabla J$ and small step size α :

$$\mathbb{E}[J(\theta + \alpha \hat{g})] \approx J(\theta) + \alpha \|\nabla J(\theta)\|_2^2 \geq J(\theta), \quad (14.9)$$

ensuring monotonic improvement in expectation.

5. The gradient does not blindly increase harmful thoughts. A remaining concern is that a thought with negative reward $r(c) < 0$ might still receive a positive advantage $A(c) > 0$ if it is simply less harmful than its peers, apparently encouraging bad reasoning. However, the gradient update does not blindly amplify such thoughts; it reallocates probability mass among them in a way that *improves the expected objective*.

First, because the advantages are defined as

$$A(c) = \frac{G}{G-1} (r(c) - \bar{r}), \quad \text{with} \quad \bar{r} = \frac{1}{G} \sum_{j=1}^G r(c^{(j)}),$$

the total $\sum_i A(c^{(i)}) = 0$. Hence, even if every reward is negative, the update is zero-sum: probability increases only for thoughts that are *less negative* than average, while it decreases for those that are worse. This shift raises the expected reward $J(\theta)$ because the expected improvement rate is

$$\frac{d}{d\tau} J(\theta(\tau)) = \text{Var}_{\pi}[r(c)] \geq 0.$$

Thus, the method performs a relative reallocation and guarantees monotonic ascent in expectation.

Second, a positive advantage $A(c) > 0$ does not deterministically increase the corresponding $r(c)$ on the next update; it increases it *in expectation*. The policy gradient on thought tokens,

$$\nabla_{\theta} \mathcal{L}_{\text{IG}} \propto -A(c) \nabla_{\theta} \log \pi_{\theta}(c),$$

acts on the relative usefulness of each thought, not its absolute reward value. Over repeated steps, the model raises the log-evidence $\log p_{\theta}(x_t | x_{<t}, c)$ for those thoughts that contribute more to prediction, thereby increasing their expected $r(c)$ relative to the slowly moving EMA baseline \bar{p}_{ϕ} .

Third, the EMA baseline prevents artificial reward inflation. Because \bar{p}_{ϕ} lags behind θ through a slow exponential moving average, any transient or spurious improvement in $r(c)$ dissipates as the baseline catches up. Sustained positive advantages arise only when the model genuinely improves predictive likelihood relative to the no-think counterfactual.

Finally, while a positive advantage can momentarily reinforce a thought whose raw reward remains negative, this update is not pathological. It simply redirects probability toward the least harmful reasoning pattern available, reducing overall loss. Over time, these relatively better thoughts typically evolve into genuinely helpful ones as their predictive evidence increases, ensuring that the training process remains stable and aligned with maximizing $J(\theta)$.

14.4.2 Numerical Illustration of Relative Advantage Updates

To make the abstract dynamics more concrete, we present a simple numerical example showing how the group-relative advantage mechanism improves the expected objective $J(\pi; r)$ even when all rewards are initially negative. Note that in this illustrative example we denote the expected reward as $J(\pi; r)$ to emphasize its dependence on the discrete policy over thoughts π and fixed rewards r_i . Conceptually, this corresponds to the same

information-gain objective $J(\theta)$ introduced in the main text, expressed here in a simplified form.

We consider four sampled thoughts c_1, c_2, c_3, c_4 with policy $\pi = [\pi_1, \pi_2, \pi_3, \pi_4]$, initialized uniformly. For each thought, the information-gain reward is

$$r_i = \log p_\theta(x_t | x_{<t}, c_i) - \log \bar{p}_\phi(x_t | x_{<t}),$$

and the group size is $G = 4$ with mean reward $\bar{r} = \frac{1}{4} \sum_i r_i$. The group-relative advantage is

$$A_i = \frac{G}{G-1} (r_i - \bar{r}) = \frac{4}{3} (r_i - \bar{r}),$$

and we assume each thought has length $|c_i| = 4$ so that per-token weight is $A_i/4$. The policy is updated by an exponentiated-gradient (replicator) step

$$\pi_{\text{new}}(i) \propto \pi_{\text{old}}(i) \exp(\eta A_i), \quad \text{with } \eta = 0.5,$$

and the expected objective is $J(\pi; r) = \sum_i \pi_i r_i$.

Although a positive advantage can momentarily reinforce a thought whose raw reward r_i is still negative, this update is not pathological. Because advantages are computed relative to the group mean, a positive A_i simply indicates that c_i is *less harmful* than its peers. Increasing its probability reallocates mass away from worse alternatives, thereby improving the expected objective J . Over subsequent updates, the model typically adapts to make these less-harmful thoughts genuinely helpful, raising r_i in expectation.

Iteration 1: all thoughts are harmful ($r_i < 0$), but one is least bad.

$$\pi^{(0)} = [0.25, 0.25, 0.25, 0.25], \quad r^{(1)} = [-0.80, -0.60, -0.50, -0.30].$$

Mean and advantages:

$$\bar{r}^{(1)} = -0.55, \quad A^{(1)} = [-0.3333, -0.0667, +0.0667, +0.3333].$$

Per-token weights: $A^{(1)}/|c| = [-0.0833, -0.0167, +0.0167, +0.0833]$. Note that c_4 has $r_4 = -0.30 < 0$ yet receives a positive advantage $A_4 = +0.3333$, so every token in c_4 gets a positive gradient. Policy update with $\eta = 0.5$ gives

$$\pi^{(1)} \propto \pi^{(0)} \odot \exp(0.5 A^{(1)}) = [0.2101, 0.2401, 0.2566, 0.2932],$$

yielding $J(\pi^{(0)}; r^{(1)}) = -0.5500$ and $J(\pi^{(1)}; r^{(1)}) = -0.5284$. This is a small but consistent improvement.

Iteration 2: dense updates improve r on c_3, c_4 .

$$r^{(2)} = [-0.80, -0.60, -0.35, -0.10],$$

$$\bar{r}^{(2)} = -0.4625,$$

$$A^{(2)} = [-0.4500, -0.1833, +0.1500, +0.4833].$$

Update:

$$\pi^{(2)} \propto \pi^{(1)} \odot \exp(0.5 A^{(2)}) = [0.1618, 0.2113, 0.2668, 0.3601].$$

Expected objective: $J(\pi^{(1)}; r^{(2)}) = -0.4313$, $J(\pi^{(2)}; r^{(2)}) = -0.3856$.

Iteration 3: the least-bad thought becomes genuinely helpful.

$$r^{(3)} = [-0.80, -0.60, -0.20, +0.05],$$

$$\bar{r}^{(3)} = -0.3875,$$

$$A^{(3)} = [-0.5500, -0.2833, +0.2500, +0.5833].$$

Policy update:

$$\pi^{(3)} \propto \pi^{(2)} \odot \exp(0.5A^{(3)}) = [0.1127, 0.1681, 0.2772, 0.4420],$$

and the expected objective improves again: $J(\pi^{(2)}; r^{(3)}) = -0.2916$, $J(\pi^{(3)}; r^{(3)}) = -0.2244$.

As seen in the above, in Iteration 1, all rewards are negative, yet c_4 (the least bad) has a positive advantage, showing how the dense loss pushes probability toward less harmful thoughts and increases J . Since rewards are tied to log-evidence, these positive gradients directly improve the corresponding $r(c)$ values, leading to less-negative and eventually positive rewards in later iterations.

14.5 Experimental Setup

RLP: We employ RLP on both base and intermediate checkpoints using diverse datasets. To facilitate this, we use [Face \(2025\)](#) as the RL training backbone and deploy training using 32 H100 80GB SXM5 GPUs for 170M to 10B tokens. We train the base models with key settings including a constant learning rate of $1e^{-6}$, a batch size of 512 and a maximum context length of 2048 tokens. Each generation step contains 512 unique prompts sampled from the dataset, and performing 16 rollouts with temperature 0.7. We set KL coefficient to 0 across all runs.

Continuous Pre-training: We continuously pretrain the $\mathcal{M}_{\text{base}}$ model using both general pretraining and specialized post-training corpus to draw comparison between pretraining and RLP training objective. For this experimentation, we use Megatron-LM ([Shoeybi et al., 2019a](#)) as the pretraining backbone and continuously train on 32 H100 80GB SXM5 GPUs for 170M to 10B tokens depending on the data size and comparison requirement. During training, we use the AdamW optimizer ([Loshchilov & Hutter, 2019](#)) with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and weight decay of 0.1. We use a 2-way tensor and pipeline parallelism to train the model. We set the maximum value of learning rate to $1e^{-6}$, minimum to $1e^{-7}$, and use a batch size of 6M tokens with a 8192 context length.

Post-Training: For supervised fine-tuning (SFT), we use the OpenThoughts3 dataset ([Guha et al., 2025](#)). We filtered examples that did not include a final answer. With this filtering scheme, the total number of samples for SFT post-training is 45,6024. For RLVR, we used the The Mathematics Aptitude Test of Heuristics (MATH) dataset ([Hendrycks et al., 2021d](#)) with 7,500 examples. This dataset includes problems from various subjects such as algebra, geometry, number theory and precalculus. We trained models in all RLVR experiments for 1 epoch with a global batch size of 1024 and used cosine annealing and an initial learning rate of $1e^{-6}$.

Prompt Given a context $x_{<t}$, we ask the model to reason about the target token x_t using the following prompt, p . `tealSystem Prompt, p "You are a continuation-and-reasoning assistant.`

You receive the prefix of a context, problem, solution, or derivation. First, briefly think between `< /think>` and `< /think>` about what should come next. Then, after `< /think>`, continue the text in the SAME style as the prefix (notation, LaTeX, tone), focusing on the next few steps rather than jumping to a final boxed answer. Do not restate the question or add meta commentary; simply continue the content."

14.6 Extended ablation details

Table S.1 reports per-task accuracies for each setting, and Fig. 10.2 provides the corresponding curves for (a) rollout count, (b) completion length, and (c) KL coefficient. Unless stated, each sweep holds the other two dimensions at the best configuration (16 rollouts, completion length 2048, $\beta = 0$).

Rollout count. Increasing G improves accuracy up to $G = 16$, where *Overall* reaches 42.17% (from 34.03%, +8.14 points). The largest taskwise lifts at $G = 16$ relative to the base are GSM8K (+22.96), MATH-500 (+13.85), MIVA (+7.20), MMLU (+6.35), and MMLU-PRO (+6.20), while GPQA is unchanged (27.51 vs 27.52). Moving from $G = 16$ to $G = 32$ slightly lowers *Overall* to 41.75 (−0.42), driven mainly by GPQA (−2.13), with other tasks nearly flat (e.g., MMLU-PRO +0.79, MMLU −0.24). This suggests diminishing returns once the group-relative estimator is already well-sampled.

Completion length. Capacity on the thought channel dominates performance. Very short completions underperform sharply: at length 64, *Overall* is 11.50 and *Math* averages 1.12. Increasing to 512 raises *Overall* to 24.65 and *Math* to 22.63. The main jump occurs between 512 and 1024 (*Overall* +14.24 to 38.89; GSM8K +28.55; MATH-500 +36.85). Extending to 2048 adds a smaller but consistent gain (*Overall* 42.17, +3.28 over 1024; *Math/Science* 48.06/36.29). Pushing to 4096 gives only a marginal change (*Overall* 42.21, +0.04; small taskwise shifts such as MMLU-PRO +0.64 and GSM8K −0.62), so 2048 is the preferred trade-off.

Model / Variant	Tasks (%)							Macro avg (%)		
	MATH500	GSM8K	AMC23	Minerva	MMLU	MMLU-Pro	GPQA	Math	Science	Overall
<i>Baseline</i>										
Qwen3-1.7B-Base	48.45	54.16	25.94	15.30	44.85	23.95	27.52	35.96	32.11	34.03
<i>Ablation: # rollouts</i>										
num_rollouts=4	59.45	74.79	33.44	21.78	50.83	28.81	26.52	47.37	35.39	41.38
num_rollouts=8	61.70	76.93	30.62	22.06	50.88	30.55	26.77	47.83	36.07	41.95
num_rollouts=16†	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	42.17
num_rollouts=32	60.45	77.26	30.94	22.29	50.96	30.94	25.38	47.74	35.76	41.75
<i>Ablation: completion length</i>										
completion_length=64	1.00	2.84	0.62	0.00	33.26	15.46	16.92	1.12	21.88	11.50
completion_length=128	1.73	3.17	0.94	0.05	29.04	13.94	12.37	1.47	18.45	9.96
completion_length=256	2.95	13.86	2.81	0.46	37.19	17.09	15.15	5.02	23.14	14.08
completion_length=512	21.35	46.58	16.25	6.34	42.27	19.82	17.93	22.63	26.67	24.65
completion_length=1024	58.20	75.13	28.80	20.47	48.36	27.74	20.31	45.65	32.14	38.89
completion_length=2048†	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	42.17
completion_length=4096	62.00	76.50	30.60	22.80	51.30	30.79	27.27	47.98	36.45	42.21
<i>Ablation: KL weight β</i>										
$\beta = 10^{-4}$	61.35	75.86	28.00	21.50	51.00	31.58	25.50	46.68	36.03	41.35
$\beta = 10^{-3}$	60.90	74.30	32.19	20.73	50.73	30.80	26.00	47.03	35.84	41.44
$\beta = 0^\dagger$	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	42.17

Table S.1: Ablations on rollout count, completion length, and KL weight β with QWEN3-1.7B-BASE. All numbers denote accuracy (%).

KL coefficient. Adding a token-level KL toward a fixed reference does not help overall. At $\beta = 10^{-4}$ and 10^{-3} , *Overall* is 41.35 and 41.44 (-0.82 and -0.73 vs $\beta = 0$). There are isolated improvements (MMLU-PRO $+1.43$ at 10^{-4} ; AMC23 $+1.88$ at 10^{-3}), but these are offset by broader declines (e.g., GSM8K -1.26 and -2.82 ; GPQA -2.01 and -1.51). The KL term also increases memory use and step time. We therefore keep $\beta = 0$ in the main recipe.

In summary, the appendix table provides the taskwise breakdown behind these trends, and the figure shows the smooth saturation with rollouts, the strong length-driven regime change between 512 and 1024 tokens, and the lack of net benefit from KL.

14.7 Additional Ablations

Model	Dataset	Math Avg@1[8]	Science Avg	Science Avg@1[4]	Average
$\mathcal{M}_{\text{base}}$	-	35.96	34.50	32.11	34.19
\mathcal{M}	Only Math	48.23	41.64	36.77	42.21
	Only Science	49.17	39.65	38.26	42.36
	Combined	49.76	42.54	37.78	43.36

Table S.2: **Ablation on math, science, and combined domains.** RLP shows particularly strong generalization in presence of multi-domain data.

Does the improvement sustain if we make Pretraining compute equivalent to RLP?

For both comparisons, the configuration for RLP remains fixed, based on a budget of $T_{\text{inp}} = 170M$ input tokens. First, we establish a baseline by continuing the pretraining of the base model on an identical 170M tokens (Base + CPT, Input Token). Second, to create a FLOP-equivalent baseline, we first approximate the total computational cost of RLP. The effective token budget, T_{flop} , can be estimated by summing the tokens used for gradient updates (T_{inp}) and the tokens processed during the rollout phase:

$$T_{\text{flop}} = (n \times l_{\text{seq}} \times bs \times \textit{iters}) + T_{\text{inp}}$$

where n is the number of rollouts per instance, l_{seq} is the sequence length, bs is the batch size and \textit{iters} is the number of steps RLP has gone through. This calculation results in an effective budget of approximately 6B tokens for our model. We therefore train a second, more powerful CPT baseline on 6B tokens (Base + CPT, Flop Usage), holding all other hyperparameters constant.

RLP resonates well in presence of multidomain data. Recent works have shown tremendous improvement in reasoning tasks, particularly in mathematics, through RLVR (Liu et al., 2025a; Luo et al., 2025a; Hu et al., 2025b). However, these methods are often tied to the complexity of queries, limiting their scalability. To draw a parallel, we evaluate RLP on NEMOTRON-CROSTHINK using different blends of math and science data. As shown in Table S.2, training only on math yields substantial math improvements, but comes at the cost of weaker generalization to science. Conversely, training only on science improves science accuracy, but underperforms in math compared to math-only training. Strikingly, combining both domains provides the best overall average, indicating that RLP is able to leverage complementary signals from multiple domains without diluting the benefits within each. This suggests that RLP not only scales beyond single-domain specialization but also thrives in multidomain settings where diverse reasoning styles reinforce one another.

Effectiveness of RLP with Scaling LLM. Validating RLP on larger, state-of-the-art model sizes is indeed essential to confirm that our gains hold as parameter counts increase. To address this, we conducted a new set of experiments applying RLP to the $\mathcal{M} = \text{QWEN3-14B-BASE}$ model and training on our general pretraining corpus (\mathcal{D}_{PT}) for 1B tokens. As shown in [Table S.3](#), RLP delivers substantial improvements even on this stronger, significantly larger baseline. Applying RLP improves the overall average from 60.66% to 65.00%, with particularly notable gains in scientific reasoning where the average score improves from 60.07% to 64.76%. These results confirm that the dense, verifier-free signal provided by RLP remains effective at scale, successfully extracting reasoning capabilities that are not fully utilized by standard pretraining alone.

Benchmarks	$\mathcal{M}_{\text{base}}$	\mathcal{M}_{RLP}
MATH500	78.81	81.15
GSM8K	90.36	94.04
AMC23	55.94	57.81
Minerva	37.96	40.26
MMLU	76.56	80.59
MMLU@1[4]	74.03	79.31
MMLU-Pro	59.20	65.53
MMLU-Pro@1[4]	54.00	61.75
GPQA	44.44	48.15
GPQA@1[4]	40.40	44.70
Math Avg	65.77	68.32
Science Avg	60.07	64.76
Science Avg@1[4]	56.14	61.92
Overall	60.66	65.00

Table S.3: RLP training with QWEN3-14B-BASE model.

How early RLP can be applied? Previously, we have confirmed that RLP can be integrated to intermediate checkpoints from last stage of pretraining. However, it is unclear whether the gains sustain if we pick a very early checkpoint for RLP training. Inspired by the finding of [Han et al.](#) and to study how early RLP can be introduced, we additionally evaluate a much earlier checkpoint. Concretely, we take a NEMOTRON-NANO-12B-V2 model trained on only 4T tokens (about 20% of the full 20T pretraining budget) and apply RLP for 1B tokens on the same pretraining corpus \mathcal{D}_{PT} . As shown in [Table S.4](#), even at this early stage, RLP is highly effective: with only 1B tokens, Math Avg more than doubles (from 21.93 to 50.14), Science Avg@1[4] improves by 6 points (from 5.69 to 11.96), and Overall Average increases by 12 points (from 12.05 to 24.08). While our strongest final results come from applying RLP later in pretraining, these findings indicate that RLP can already yield large gains when the model has seen only a small fraction of the standard pretraining budget.

FLOP matched comparison between RLP and RPT. We would like to clarify that even though RLP can be theoretically applied to tokens at every position in the document, in practice we only apply it for one token per document. This token is selected randomly and not through any criteria as in the case of RPT. For the experiments in [Table 10.3](#), we have matched the number of input tokens for both RLP and RPT settings (we train both methods for one epoch of the same documents). But we want to highlight that the number of target tokens for which reward is calculated is much larger for RPT compared to RLP (since we don’t apply the RLP reward to every token in the document). Hence, the setting in [Table 10.3](#) is in favor of RPT. Additionally, we don’t include the compute needed to pre-select tokens using an external LLM for RPT.

To directly address the head to head flop matched comparison, we run a controlled experiment using Nemotron-CrossThink data. We deploy both RLP and RPT for only one epoch on the same data, i.e., the number of target tokens for which reward is calculated is similar in both cases. RLP achieves a 16.23% relative improvement in Overall Avg and consistently outperforms RPT on both math and science aggregates. These results confirm that the gains

in Table 3 are not an artifact of mismatched settings. Even under stricter, target-matched conditions, RLP provides stronger and more general improvements.

As shown in Table 10.3, RLP achieves a 16.23% relative improvement in Overall Avg and consistently outperforms RPT on both math and science aggregates. These results confirm that the gains in Table 3 are not an artifact of mismatched settings. Even under stricter, target-matched conditions, RLP provides stronger and more general improvements.

Continuous pretraining with longer context length.

Qwen3-1.7B-Base is indeed eventually extended to a 32K context window, but as described in the Qwen3 technical report, this happens only in a third long-context stage after the model has already been pretrained for 30T+ tokens at a much shorter context (4,096 tokens) and then further trained on knowledge-intensive data. Our CPT experiments are conceptually closer to these first two stages; we continue pretraining on our pretraining mixture (\mathcal{D}_{PT}), which consists almost entirely of relatively short documents without long-range dependencies. In this regime, substantially increasing the context length does not obviously provide additional learning signal, but does change the optimization landscape and the effective batch and gradient statistics.

To evaluate the effect of longer context length, we conduct an additional controlled experiment where we keep all CPT hyperparameters fixed and only increased the context length from 8K to 32K. The resulting model, denoted ($\mathcal{M}_{CPT}(32K)$), is compared to our original ($\mathcal{M}_{CPT}(8K)$) in Table S.5. The result suggests that for our pretraining corpus (\mathcal{D}_{PT}), which rarely contains long documents that would actually utilize a 32K window, the 8K context configuration is at least as strong as, and in practice strictly better than, a 32K context configuration under matched compute and hyperparameters. Therefore, while we agree that context length is an important design choice, in our specific setup, using 8K rather than 32K does not weaken the CPT baseline; if anything, the longer context hurts optimization without yielding downstream benefits. Importantly, all comparisons between RLP and CPT are made against the stronger 8K CPT configuration.

Effect of EMA over RLP training. In RLP, the EMA baseline (\bar{p}_ϕ) acts as a dynamic no-think counterfactual, providing a reference log-likelihood for each next token. The decay

Benchmarks	\mathcal{M}_{base}	\mathcal{M}_{RLP}
MATH500	30.15	62.38
GSM8K	29.56	81.42
AMC23	22.81	37.81
Minerva	5.19	18.93
MMLU	11.59	13.10
MMLU@1[4]	8.73	20.68
MMLU-Pro	4.93	6.11
MMLU-Pro@1[4]	2.66	7.50
GPQA	9.10	11.20
GPQA@1[4]	5.68	7.70
Math Avg	21.93	50.14
Science Avg	8.54	10.14
Science Avg@1[4]	5.69	11.96
Overall	12.05	24.08

Table S.4: Comparison of NEMOTRON-NANO-12B-V2 4T Base and Base+RLP across benchmarks.

Benchmarks	$\mathcal{M}_{CPT}(8K)$	$\mathcal{M}_{CPT}(32K)$
AIME25	3.96	3.33
MATH500	57.52	51.80
GSM8K	72.85	60.44
AMC23	31.25	25.00
Minerva	19.03	17.46
MMLU	41.95	42.19
MMLU@1[4]	40.00	40.55
MMLU-Pro	27.81	27.08
MMLU-Pro@1[4]	24.61	22.87
GPQA	26.26	25.76
GPQA@1[4]	24.75	24.21
Math Avg	36.92	31.61
Science Avg	32.01	31.68
Science Avg@1[4]	29.79	29.21
Overall	32.90	30.83

Table S.5: Comparison of CPT models with 8K vs 32K context length.

rate τ controls comparison difficulty: if τ is too low, the baseline updates too quickly and the reward collapses toward zero; if too high, it becomes stale and yields artificially easy gains.

To justify our choice of τ , we ran a sensitivity study on Qwen3-1.7B-Base with $\tau \in \{0.99, 0.995, 0.999, 0.9995\}$. As shown in Table S.6, performance forms a bell-shaped curve with a clear peak at $\tau = 0.999$.

Model	τ	Math Avg	Science Avg	Overall Avg
$\mathcal{M}_{\text{base}}$	N/A	35.96	32.11	34.03
	0.99	45.20	36.31	38.82
\mathcal{M}_{RLP}	0.995	45.18	37.36	39.21
	0.999	45.98	37.38	39.54
	0.9995	45.64	36.84	39.20

Table S.6: Effect of temperature τ on performance. Best result highlighted.

Across this range, training remained stable and we did not observe divergent or unstable behavior in any of our runs. Concerns that the model could “game” the objective by degrading the baseline do not manifest because the baseline is updated only via the EMA of the student parameters: for the baseline to degrade, the student must degrade first, which is immediately penalized through the primary reward term $\log p_\theta$. Thus the EMA baseline provides a stable, meaningful measure of information gain.

Wall-clock time of RLP training versus SFT. We conduct a direct comparison using 32 H100 GPUs with a global batch size of 512 and a 32K context length. As shown in the table below, RLP incurs an expected overhead due to the generation phase ($G = 16$ rollouts). While SFT, which has a similar computational profile to standard Continuous Pretraining (CPT), achieves a throughput of 92.34 samples/s (approx. 5.5s per step), RLP operates at 41.07 samples/s (approx. 12.5s per step). This results in a per-step slowdown factor of roughly $2.25\times$.

Method	Batch Size	Rollouts (G)	Time/Step (s)	Throughput (samples/s)	Relative Speed
SFT	512	N/A	5.54	92.34	1.00 \times
RLP	512	16	12.47	41.07	0.44 \times

Table S.7: Comparison of SFT and RLP training efficiency.

However, this per-step cost must be viewed in the context of convergence efficiency and total compute. While RLP is $2.25\times$ slower per iteration than SFT/CPT, it is drastically more data-efficient. As detailed in Table 10.4, RLP achieves an overall average accuracy of 43.36% on the Nemotron-Crosstink dataset using only 170M tokens. In contrast, the FLOP-matched CPT baseline required processing 6B tokens (roughly $35\times$ more data) to account for the compute difference, yet only reached an accuracy of 35.60%. Thus, while RLP processes tokens slower due to rollouts, the dense reward signal extracts significantly more reasoning capability per FLOP, yielding a performance margin (+7.76%) that standard training cannot replicate even with substantially higher data volume.

Final perplexity after post-training. We confirm that the model’s perplexity on ordinary tokens does not degrade; in fact, it significantly improves. Unlike standard RLHF, where optimizing for an external reward often causes the model distribution to drift away from natural language, our reward signal is the log-likelihood of the next token itself. Therefore, by definition, RLP is optimizing for prediction accuracy.

Model	Nemotron-CrossThink PPL ↓	Nemotron-CrossThink NLL ↓	Wikitext-103 PPL ↓	Wikitext-103 NLL ↓
$\mathcal{M}_{\text{base}}$ (Qwen-1.7B)	2.91	1.06	5.83	1.77
\mathcal{M}_{RLP} (Ours)	2.36	0.86	4.48	1.50

Table S.8: Perplexity and NLL comparison on Nemotron CrossThink and Wikitext-103.

Mathematically, maximizing the RLP reward is equivalent to minimizing the cross-entropy of the reasoned predictor against the data distribution (Proposition 1). As shown in the table below, our empirical results confirm this theoretical guarantee: \mathcal{M}_{RLP} achieves consistently lower Perplexity (PPL) and Negative Log-Likelihood (NLL) compared to the base model. Crucially, this improvement holds for both the reasoning-intensive Nemotron CrossThink dataset and the general-domain Wikitext-103 benchmark, demonstrating that the “thoughts” generated by the model successfully compress information to better predict ordinary text.

Computational Cost and FLOP Analysis of RLP. A potential concern in comparing RLP against CPT is the perceived computational burden of autoregressively generating long reasoning traces. This would indeed be prohibitive if the rollout policy were applied at every token position in a sequence. In practice, however, RLP is applied to only *one randomly sampled token per sequence*, which dramatically reduces the computational burden. Instead of scaling with $L_{\text{doc}} \times L_{\text{CoT}}$, the rollout cost scales with $1 \times L_{\text{CoT}}$ per sequence. This design choice makes RLP computationally feasible and allows us to interleave it with standard training efficiently. In addition, autoregressive generation involves a bottleneck compared to parallel processing. We agree that this affects wall-clock time due to memory bandwidth constraints, but it does not incorrectly skew the *FLOP* calculation used for the baselines.

In Appendix 14.7, we calculated the FLOP-equivalent budget by summing the tokens used for gradient updates and the tokens generated during rollouts. We compared RLP (170M input tokens) against a CPT baseline trained on 6B tokens. This $35\times$ increase in data for the baseline is a rigorous upper bound for two reasons:

- **Operation Count:** The FLOPs of a forward pass for generating one token is approximately $2N$ (where N is parameter count). The cost of training on one token (forward + backward) is approximately $6N$. By equating one generated token to one trained token in our FLOP calculation, we are effectively penalizing RLP (counting generation as $3\times$ more expensive than it theoretically is in terms of FLOPs).
- **Total Compute:** Even with the overhead of 16 rollouts of length 2048 per document, the total floating-point operations performed by RLP on 170M documents are comparable to (or less than) performing standard forward/backward passes on the 6B tokens used in the $\mathcal{M}_{\text{CPT}}[6\text{B}]$ baseline.

While autoregressive generation is indeed slower in terms of wall-clock time, the purpose of the baseline is to compare compute efficiency. RLP applied to a single token per document

is highly efficient, and our $M_{\text{CPT}}[6\text{B}]$ baseline represents a compute-matched, which RLP still outperforms significantly (Overall Avg 42.13% vs 38.04%).

On Self-Referentiality and the Semantics of the RLP Reward. A natural concern for any method that leverages model-internal signals is whether the learning dynamics risk becoming self-referential—rewarding increases in internal confidence rather than genuine improvements in correctness or reasoning ability. In RLP, however, the reward structure is explicitly grounded in the data rather than in unconstrained model self-agreement.

- **Reward is anchored to ground-truth tokens.** For each sampled position, the reward

$$r(c_t) = \log p_\theta(x_t | x_{<t}, c_t) - \log \bar{p}_\phi(x_t | x_{<t})$$

is defined with respect to the ground-truth next token x_t from the corpus. Proposition 1 shows that, in expectation over $x_t \sim p^*(\cdot | x_{<t})$, this reward equals the reduction in cross-entropy achieved by conditioning on the thought c_t . A thought therefore receives positive reward only if it moves probability mass *toward* the correct continuation in the true data distribution. Increased confidence on an incorrect continuation strictly decreases the reward. This prevents the model from benefiting by simply inflating logit magnitudes or reinforcing patterns unrelated to accuracy.

- **EMA baseline prevents degenerate self-consistency loops.** RLP compares each thought-conditioned prediction to an exponential moving average (EMA) baseline \bar{p}_ϕ evaluated on the same context and same ground-truth token. If the current parameters shift toward patterns that improve internal consistency but harm prediction of the observed token, the relative likelihood under p_θ falls and the corresponding thought receives a negative advantage. Group-relative normalization and advantage clipping further ensure that thoughts cannot win reward by global logit scaling alone; only thoughts that contribute meaningful information about the next token outperform the EMA teacher in expectation.
- **External evaluations validate correctness rather than internal consistency.** The most important empirical question is whether internal information gain translates into better reasoning on verifiable tasks. Across GSM8K, MATH500, MMLU-Pro, GPQA, and other benchmarks with objectively checkable answers, RLP-trained models consistently outperform both the base model and compute-matched continuous-pretraining baselines—even when the latter consume substantially more training tokens at equal FLOPs. Notably, these gains persist after a strong post-training pipeline involving SFT and RLVR with external verifiers. If RLP were primarily amplifying internal confidence without improving correctness, these advantages would be expected to collapse or become fragile under verifier supervision. Instead, RLP-initialized models remain ahead, particularly on reasoning-heavy domains, indicating that the learned thoughts encode genuinely useful information and not merely self-reinforcing patterns.

Overall, the formulation of the RLP reward ensures that the model is optimized for meaningful reductions in predictive error on the underlying data distribution, while empirical evidence confirms that these internal information gains translate into improved external reasoning performance.

Generalizability across Architectures and Data Distributions To rigorously assess the universality of our approach, we evaluated RLP on two models chosen specifically for their

significant divergence in both architectural design and data provenance: Qwen3-1.7B-Base and Nemotron-Nano-12B-V2. These distinct settings demonstrate that RLP is not limited to a single model family or training recipe.

- **Architectural Heterogeneity:** The models represent fundamentally different backbone architectures. Qwen3-1.7B-Base utilizes a standard, pure Transformer architecture. In contrast, Nemotron-Nano-12B-V2 is a *Hybrid Mamba2-Transformer*, which integrates State Space Models (SSM) with attention layers and employs a distinct tokenizer and training recipe. The fact that RLP translates effectively to this hybrid architecture—driving an increase in overall average accuracy from 42.81% to 61.32%—provides strong evidence that the method is architecture-agnostic.
- **Data Lineage Independence:** The models also rely on distinct pretraining distributions. Nemotron-Nano-12B-V2 was trained from scratch on a massive mixture of raw public data (including Web Crawl, arXiv, and OpenWebMath) and synthetic data generated by a diverse ensemble of teacher models. As detailed in recent technical reports, this ensemble extends well beyond the Qwen family to include DeepSeek-R1 (DeepSeek-AI, 2025a), Mixtral-8x22B-v0.1 (Jiang et al., 2024), DeepSeek-V3 (DeepSeek-AI, 2025b), Phi-4 (Abdin et al., 2024a), and Nemotron-4-340B (Nvidia et al., 2024).

By validating RLP on a Hybrid Mamba model trained on a multi-source ensemble distribution, distinct from the pure Transformer Qwen baseline, we confirm the method’s scalability and robustness across disparate experimental environments.

Analysis of thought traces. We examine the model’s internal thought traces to understand how RLP shapes reasoning. A manual analysis of 50 randomly sampled traces reveals consistently high-quality reasoning, characterized by grammatical correctness and strong contextual relevance, confirming that the information-gain objective effectively suppresses unproductive generation. We observe three qualitative properties: (1) *focused, context-aware reasoning*, where thoughts orient the model to the specific logical step required by the context; (2) *continuation-style* formatting, acting as internal scratch work that mirrors the prefix’s tone and notation without meta-commentary; and (3) *utility-driven anticipation*, where thoughts predict structural moves (e.g., upcoming rules or algebraic steps) to maximize the likelihood of the ground-truth continuation. An illustrative example follows.

tealPrefix “West Virginia students invited to enter Ornament Competition CHARLESTON, W.Va. (AP) — West Virginia students in kindergarten through 12th grade may enter the First Lady Student Ornament Competition this fall. First lady Cathy Justice is asking all students to participate in the 18th annual event. It is open to students in public and private schools as well as those who are home-schooled. Students are asked to create a “Nutcracker”-themed ornament for a tree to be displayed at the Culture Center in Charleston during the holidays, according to a news release from Gov. Jim Justice’s office. There will be four divisions according to grade, and a winning class will be chosen from each division. The winning ornaments will be donated in January to the West Virginia State Museum. The four winning classes will”

tealThought Trace "The article is outlining contest logistics. The next sentence will likely add a specific detail such as what the winning classes receive, how the ornaments will be

Benchmark	OmniMath	OpenThoughts	Nemotron-CrossThink	ACAD	Math-Text	Web-Crawl
MATH500	57.95	59.55	62.65	59.75	60.03	61.58
GSM8K	74.82	74.80	79.97	76.10	75.95	76.48
AMC23	32.50	32.81	30.94	31.88	34.38	35.00
Minerva	20.63	20.96	25.46	22.98	21.92	22.43
MMLU	55.72	55.84	56.72	56.35	56.02	55.93
MMLU@1[4]	50.85	50.84	52.11	51.51	50.72	50.11
MMLU-Pro	35.81	34.55	38.58	35.11	36.06	37.02
MMLU-Pro@1[4]	31.47	30.57	35.10	31.06	31.86	32.43
GPQA	29.29	25.76	32.32	30.30	29.29	29.29
GPQA@1[4]	30.30	27.27	26.14	28.03	26.39	27.78
Math Avg	46.48	47.03	49.76	47.68	48.07	48.87
Science Avg	40.27	38.72	42.54	40.59	40.46	40.75
Science Avg@1[4]	37.54	36.23	37.78	36.87	36.32	36.77
Overall	41.43	40.66	43.36	41.71	41.62	42.13

Table S.9: Quantitative benchmarks for QWEN3-1.7B-BASE, showing the impact of RLP on different data blends. Shaded columns indicate general pretraining corpus.

displayed, or other submission guidelines. Maintain the neutral news tone and extend the informational structure already established."

14.8 Data Blend Extended Results

To further examine whether RLP learns transferable reasoning beyond narrowly curated datasets, we evaluate it across a broad spectrum of corpora spanning both structured reasoning data and open-ended pretraining distributions. All experiments start from QWEN3-1.7B-BASE and apply RLP for 200 steps, consuming 170M input tokens, while keeping all other training settings fixed.

We consider two primary corpus families. The first consists of SFT-style reasoning datasets, including OmniMath (Gao et al., 2024a), OpenThoughts (Guha et al., 2025), and Nemotron-CrossThink (Akter et al., 2025). These datasets contain structured question–answer pairs with explicit reasoning content and represent the typical setting where reinforcement-based methods are expected to perform well. The second family consists of general-purpose pretraining corpora, including academic papers (ACAD), math textbooks (Math-Text), and open-ended web crawl data. These datasets are not curated specifically for reasoning and more closely resemble large-scale pretraining mixtures.

Table 10.4 shows that RLP consistently improves over the base model across all corpus types. Importantly, the gains are not confined to math-centric SFT data. While the strongest improvements occur on Nemotron-CrossThink within the SFT family, substantial gains are also observed when training on purely general corpora such as academic papers and web crawl data. This demonstrates that RLP does not depend on carefully constructed reasoning traces. Instead, it extracts a transferable reasoning signal even from heterogeneous, weakly structured text.

A notable contrast emerges when compared with prior RL-based approaches that report improvements concentrated in high-quality math data and limited transfer to broader domains. In our experiments, models trained with mixed-domain or open-ended corpora simultaneously improve math, science, and professional benchmarks. There is no evidence

of domain-specific overfitting or degradation on math when incorporating diverse data. Rather, diversity appears to strengthen general reasoning performance.

Table S.9 provides a task-level breakdown. Across MATH500, GSM8K, MMLU, MMLU-Pro, GPQA, and related metrics, improvements are observed regardless of whether the underlying training corpus is structured SFT data or general pretraining text. Even web-scale crawl data yields competitive math and science averages, suggesting that RLP leverages latent reasoning patterns embedded in broad distributions.

Overall, these results support three conclusions. First, RLP scales across corpus families without requiring specialized reasoning datasets. Second, it exhibits genuine cross-domain transfer rather than narrow task adaptation. Third, data diversity amplifies the learned reasoning signal instead of diluting it. Together, this positions RLP as a domain-agnostic pretraining augmentation that enhances both reasoning robustness and general benchmark accuracy.