Scalable Alignment of Large Language Models Towards Truth Seeking, Complex Reasoning, and Human Values

Zhiqing Sun

CMU-LTI-25-006

April 24, 2025

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15123

Thesis Committee:

Yiming Yang (Chair)	Carnegie Mellon University
Lei Li	Carnegie Mellon University
Sean Welleck	Carnegie Mellon University
Denny Zhou	Google DeepMind

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

Keywords: large language models, AI reasoning, AI alignment, scalable oversight

Abstract

The exponential advancement in Large Language Models (LLMs) and reasoning-powered AI agents, exemplified by GPT-4 and OpenAI Deep Research, has accelerated the timeline toward Artificial General Intelligence (AGI), with capabilities expanding at an unprecedented rate. As we stand at the threshold of potentially achieving AGI in the near future, the challenge of alignment—ensuring these systems remain truthful, capable of sophisticated reasoning, and aligned with human values—has become increasingly critical.

This thesis proposes novel methodologies to address fundamental alignment challenges for systems approaching superhuman capabilities. Extending beyond conventional paradigms such as Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), we develop scalable alignment mechanisms through our Principle-Driven Alignment methodology. Implemented within a reinforcement learning from AI feedback (RLAIF) framework, this approach demonstrates significant improvements in maintaining system reliability under capability scaling. To mitigate factual inconsistencies in generation, we introduce Recitation Augmentation and Factually Augmented RLHF, which demonstrate robust performance on large language and multimodal models. The proposed Easy-to-Hard Generalization framework provides a systematic approach for preserving alignment by leveraging the insight that models can more reliably evaluate solutions than generate them, enabling supervision of complex reasoning tasks through reward models trained on simpler problems. Additionally, we proposed Lean-STaR, a framework that improves theorem-proving performance by guiding models to generate informal thoughts before formal solutions, demonstrating the effectiveness of Chain-of-Thought reasoning in enhancing autonomous decision-making capabilities while providing greater transparency of model reasoning processes.

This research contributes to a critical area of AI development by establishing rigorous frameworks for maintaining alignment as systems become increasingly capable. Our findings demonstrate the effectiveness of these approaches in creating AI systems that are aligned with fundamental human values while preserving performance reliability. These frameworks provide a foundation for scalable solutions that will shape the future development of advanced AI systems.

Acknowledgments

When I look back at my time at Carnegie Mellon University, I'm struck by how much has changed since I first arrived in 2019. I came in both excited and unsure – my English felt shaky, and I'd barely set foot in the United States before. Large language models were still young, and I never would have guessed how quickly they'd become central to almost every AI conversation. These six years turned out to be a whirlwind of learning, exploration, and resilience, especially with the pandemic affecting more than half of my PhD journey. I'm grateful to everyone who walked alongside me, offering wisdom and friendship through every twist and turn.

First and foremost, my heartfelt thanks go to my advisor, Yiming Yang. I originally worked on word segmentation and machine translation, but she encouraged me to explore entirely new directions – first combinatorial optimization in my third year, and then large language models again in my fourth. Throughout it all, Yiming taught me how to write clearly, present with confidence, and think more critically about research. Her patience, encouragement, and generosity were the cornerstones of my growth, and I feel incredibly fortunate to have had her as a mentor.

I'm also deeply thankful to my other committee members, each of whom has left a significant imprint on this work. In particular, Denny Zhou deserves a heartfelt mention. Back when I was an undergrad in 2019, Denny interviewed me (alongside Lihong) for an internship at Google Brain in Beijing, and during that time he mentored me remotely. It was my first deep dive into BERT and TPU-based experimentation – an eye-opening experience that made me appreciate the sheer power of large language models. Several years later, I finally got to work with Denny in person in Seattle. That was right when Chain-of-Thought reasoning was taking off, and it reignited my passion for LLMs in a way I hadn't felt since my initial discovery of them. I'm grateful for his trust, his guidance, and for giving me not just one but two incredible opportunities to learn from him. To Lei Li and Sean Welleck, thank you for your generous feedback, thoughtful questions, and encouragement. Sean's work on AI4MATH, in particular, helped expand my perspective on mathematical reasoning with large language models, influencing many of my own research directions.

Beyond Carnegie Mellon, I owe a great deal to mentors who guided me at different points in my journey. Chuang and Yikang at the MIT-IBM Watson AI Lab welcomed me into a space where I could experiment with Self-Align, SALMON, Fact-RLHF, and E2H using dedicated compute resources and invaluable feedback. In fact, the majority of the work presented in this thesis was developed during my time at the lab, and I'm deeply grateful for their support and mentorship throughout. I will always appreciate Chuang's sweeping high-level vision and Yikang's meticulous eye for technical details. I'm also indebted to my undergraduate mentors at Peking University – Zhi-Hong, Jian, Jian-Yun, and Dongdong – who introduced me to NLP and planted the seeds of my passion for AI research. Their early encouragement was crucial in shaping my path forward. I am likewise thankful to Sachin at AI2 for hosting me during a spring internship in Seattle.

A special thanks to Liam Fedus, who trusted me with the opportunity to join OpenAI's Post Training Core Modeling Team – my first full-time position – while I was still finalizing my PhD. I'm deeply grateful for his trust, understanding, and enduring positivity, all of which have profoundly influenced me.

I would be remiss not to mention my wonderful friends and collaborators at CMU and beyond. Thank you, in particular, to Shengcao, Sheng, Zhengbao, Frank, Shuyan, Luyu, and Yue for their fresh ideas, lively conversations, and moral support throughout the years. I am equally grateful to my labmates in Yiming's group such as Zihang, Guokun, Aman, Donghan, Ruohong, Shengyu, and Shanda. To my mentees – Ruizhong, Junwei, Zhiying, Haohan, and Yangzhen – you have taught me just as much as I hope I have taught you. Watching you grow has been one of the most rewarding parts of my PhD experience.

I'm deeply grateful to my family for their unwavering support, even when separated by time zones and pandemic travel restrictions. To my parents, Qiongfen and Lijun, thank you for the constant encouragement and for cheering me on from afar. And to my wife, Mai, your kindness and patience helped me navigate the ups and downs of research; I couldn't have done this without you.

There are countless others – friends, teachers, colleagues – who have shaped my life at CMU and beyond, and I wish I could name all of you. Thank you for making these six years such a rewarding adventure. I hope this thesis reflects not just my own work, but also the inspiration, advice, and moral support I've received from so many remarkable people along the way.

Contents

1 Introduction **1**

1.1 Background and Motivations 2
1.2 Alignment for Human Values Preserving 4
1.3 Alignment for Truth Seeking 6
1.4 Alignment for Complex Reasoning 8
1.5 Publications and Research Contributions 9

I Aligning Language Models Towards Human Values **11**

2 Self-Alignment of Language Models with Principle-Driven Prompting **13**

2.1 Introduction 13
2.2 Methodology Overview 16
2.3 Topic-Guided Red-Teaming Self-Instruct 17
2.4 Principle-Driven Self-Alignment 19
2.5 Principle Engraving 20
2.6 Verbose Cloning 21
2.7 Experiments 21
2.8 Conclusion & Discussion 25

3 Self-Alignment of Language Models with RLAIF & Principle-Following Reward Models **27**

3.1 Introduction 28
3.2 Reinforcement Learning with Preference Modeling 29
3.3 Principle-Driven Preference Modeling 31
3.4 RL with Principle-following Reward Models 34

3.5 Experiments . . . 36
3.6 Conclusion & Discussion . . . 39

II Aligning Language Models Towards Truth Seeking 41

4 Improving Truthfulness of Language Models with Recitation Augmentations 43
4.1 Introduction . . . 43
4.2 Methodology Overview . . . 45
4.3 Prompt-based Recite-and-Answer for Question-Answering . . . 46
4.4 Passage Hint-based Diversified Recitation with Fine-Tuning . . . 48
4.5 Experiments . . . 49
4.6 Conclusion & Discussion . . . 54

5 Aligning Multimodal Models with Factually Augmented RLHF 57
5.1 Introduction . . . 57
5.2 Methodology Overview . . . 61
5.3 Multimodal RLHF . . . 61
5.4 Factually Augmented RLHF (Fact-RLHF) . . . 64
5.5 Experiments . . . 65
5.6 Discussions & Limitations . . . 69
5.7 Conclusion . . . 70

III Aligning Language Models Towards Complex Reasoning 71

6 Easy-to-Hard Generalization: Scalable Alignment Beyond Human Supervision 73
6.1 Introduction . . . 73
6.2 Related Work . . . 76
6.3 Methodology . . . 78
6.4 Main Results . . . 82
6.5 Conclusion . . . 87

7 Lean-STaR: Learning to Interleave Thinking and Proving 89
7.1 Introduction . . . 89
7.2 Related Work . . . 91

7.3 Our Method: Lean-STaR . . . 93
7.4 Experiments . . . 97
7.5 Conclusion & Limitations . . . 101

IV Conclusion **103**

8 Conclusion **105**

V Appendices **107**
8.1 Appendix of Chapter 2 . . . 109
8.2 Appendix of Chapter 3 . . . 109
8.3 Appendix of Chapter 4 . . . 109
8.4 Appendix of Chapter 5 . . . 109
8.5 Appendix of Chapter 6 . . . 109
8.6 Appendix of Chapter 7 . . . 109

Bibliography **111**

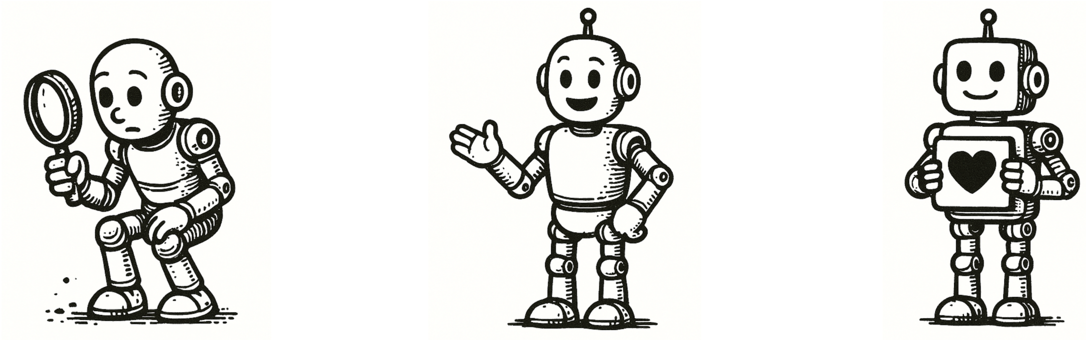
Chapter 1

Introduction

The field of Artificial Intelligence stands at a pivotal moment in its evolution, marked by exponential advancements in Large Language Models (LLMs) and reasoning-powered AI agents. The emergence of systems like GPT-4 and OpenAI Deep Research has demonstrated unprecedented capabilities in understanding, reasoning, and decision-making, accelerating the trajectory toward Artificial General Intelligence (AGI) [23, 36, 140, 141, 191]. As these systems exhibit increasingly sophisticated abilities across diverse domains, a critical challenge emerges: ensuring that AI systems remain truthful, capable of robust decision-making, and fundamentally aligned with human values while their capabilities continue to expand at an unprecedented rate.

Traditional alignment approaches, primarily Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), face significant limitations in scaling with advancing AI capabilities [144, 178]. The core challenge lies in their heavy dependence on human oversight and annotation, a constraint that becomes particularly problematic as we approach systems that may surpass human-level performance in various domains [24]. This limitation underscores the urgent need for novel alignment methodologies that can scale effectively with increasingly capable AI systems.

This thesis presents novel frameworks for maintaining alignment as AI capabilities approach and potentially surpass human-level performance. Our research extends beyond traditional human oversight paradigms [11, 16, 17, 38, 141, 144, 146], introducing scalable solutions across multiple dimensions of alignment. In Chapter 2, we demonstrate that principle-driven in-context alignment can achieve comparable performance to conventional SFT/RLHF approaches. Chapter 3 establishes RLAIF as a viable replacement for RLHF, enhancing both alignment and capabilities. Our work in Chapter 4 pioneers the use of in-context alignment for reducing hallucinations in LLM outputs, while Chapter 5 extends this to the multimodal domain through factu-



Honest (Truth-Seeking)
Recitation-Augmented Generation (Sun et al. [179], **ICLR 2023**); Factually Augmented Reward Models (Sun et al. [180], **ACL Findings 2024**)

Helpful (Reasoning)
Easy-to-Hard Generalization (Sun et al. [184], **NeurIPS 2024**); Lean-STaR (Lin et al. [115], **ICLR 2025**); Inference Scaling Laws (Wu et al. [213], **ICLR 2025**)

Harmless
Principle-Driven Self-Alignment (Sun et al. [182], **NeurIPS 2023**); Instructable Reward Model (Sun et al. [181], **ICLR 2024**)

Figure 1.1: Roadmap of this thesis: aligning AI to be **honest**, **helpful**, and **harmless**.

ally augmented reward models. Chapter 6 introduces frameworks for enhancing autonomous reasoning capabilities through easy-to-hard generalization, while Chapter 7 develops Lean-STaR to improve theorem-proving performance through the integration of informal Chain-of-Thought reasoning processes.

As we approach the potential development of AGI, the significance of robust alignment methodologies becomes increasingly critical. Our research anticipates this future, developing frameworks that not only address current alignment challenges but also establish foundations for maintaining control and beneficial outcomes as AI capabilities continue to advance. These contributions aim to shape the development of AI systems that can reliably exceed human performance while remaining fundamentally aligned with human values and objectives, ultimately contributing to the safe and beneficial advancement of artificial intelligence in society.

1.1 Background and Motivations

Open Challenges: Alignment with Human Imitation and Preferences A significant challenge in AI alignment [64] is the dependency on human-annotated data. Supervised Fine-Tuning (SFT) is a method that employs imitation learning with human demonstrations, using sources like existing NLP datasets [40, 165, 204, 206] and specially crafted instructions [49, 97, 144, 238]. Building on SFT, the Reinforcement Learning from Human Feedback (RLHF)

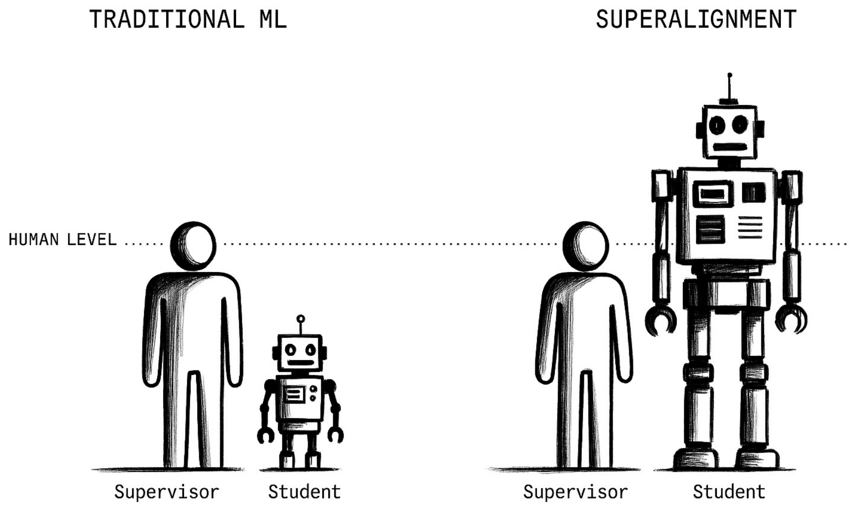


Figure 1.2: An illustration comparing traditional machine learning, where the student model remains below human level, to superalignment, where a stronger AI student is trained under scalable oversight.

paradigm [16, 38, 144, 178, 194] involves training a reward model using online human preferences to further refine the SFT-trained models [105]. GPT-4 [141] represents a significant advancement in this area, integrating a post-training alignment process to enhance factuality and adherence to desired behavior while addressing potential risks.

However, acquiring high-quality human annotations, including consistent response demonstrations and in-distribution preferences, has emerged as a significant bottleneck, because the acquisition process could be costly and raises concerns about quality, reliability, diversity, creativity, self-consistency, and the potential for undesirable biases [97, 197, 203]. Additionally, there is a concern that the current formats of response demonstrations and preferences might not generalize effectively to more complex tasks in the future.

Our Approaches: Scalable Alignment from Scratch To address the aforementioned limitations of current AI alignment methods, a new paradigm is clearly needed to support **"scalable alignment"** of AI models [22, 133]. This paradigm seeks to develop alignment methodologies that are not only effective in aligning large language models (LLMs) with human values and intentions but also efficient and scalable in their application. The essence of scalable alignment lies in its capacity to adapt and remain effective as models grow in complexity and capability, ensuring that these advanced AI systems continue to adhere to human ethics, truthfulness, and helpfulness.

In this context, **scalable oversight** (Fig. 1.2) emerges as a crucial component of scalable alignment [22, 159]. Scalable oversight techniques are designed to enhance the ability of humans to supervise models, especially as they become more complex. They include methods where models critique the outputs of other models [83, 166] and techniques for decomposing problems into simpler subproblems [37, 105, 114, 240]. Unlike approaches that solely focus on human supervision, scalable oversight prepares models to perform well even in settings where human supervision is limited or impractical. This involves developing methodologies for self-regulation and self-critique in AI models, thus ensuring their alignment with human values in complex scenarios where direct human oversight might be challenging.

In this thesis, we focus on the problem of aligning LLMs **"from scratch"**, that is, we aim to align pre-trained large language models without directly distilling from any well-aligned AI models like `Chat`GPT [140] or GPT-`4` [141]. This is markedly different from some works where the primary focus is on cloning the capabilities or well-aligned behavior from proprietary models to smaller open-source models [34, 189], which has notable drawbacks [71].

Research Dimensions: Harmless, Honest, and Helpful In tackling AI alignment, specifically for LLMs, we confront a triad of critical challenges: **harmlessness**, **honesty**, and **helpfulness** [11]. Each of these facets presents its own set of unique obstacles and considerations. Harmlessness involves aligning AI actions with human-defined ethical principles, ensuring that AI behavior does not cause unintended harm. Honesty or truthfulness is about maintaining the integrity and accuracy of information processed and generated by these models. Lastly, helpfulness pertains to the AI's ability to provide complex reasoning and problem-solving capabilities with minimal errors. The overarching goal is to develop alignment strategies that holistically address these three pillars, leading to AI systems that are not only powerful and sophisticated but also safe, trustworthy, and beneficial to humanity. Our roadmap is summarized in Fig. 1.1.

1.2 Alignment for Human Values Preserving

In the pursuit of aligning increasingly capable AI systems with human values, the concept of **"principle-driven self-alignment"** emerges as a crucial strategy [181, 182]. This approach aims to equip AI with the ability to align itself with a small set of human-defined principles, thereby minimizing the need for extensive human supervision. The primary objective is to control AI behavior more precisely while significantly reducing the dependency on human-generated annotations.

The vision of self-alignment is inspired by the notion of creating a small set of general, yet powerful, principles that AI systems can internalize and adhere to [65, 67]. This idea is reminiscent of Isaac Asimov's famous Three Laws of Robotics [10], which provided a foundational set of rules designed to govern the behavior of robots in a manner that is safe and beneficial to humans. Similarly, in our context, we aim to guide AI systems in their decision-making processes with only a concise and comprehensive set of principles. These principles would serve as a cornerstone for AI behavior, ensuring that the systems operate within an ethical and value-aligned framework, irrespective of the complexity or context of the tasks they are performing.

Principle-Driven Prompting Our initial SELF-ALIGN strategy (Chapter 2) involves implementing a prompting-based approach, wherein we introduce a compact set of 16 human-authored principles. These principles are articulated in English and focus on guiding the AI system to generate responses that are helpful, ethical, and reliable [11, 17]. They act as a foundational framework for the AI model, dictating the acceptable standards and behaviors when producing answers.

To operationalize these principles, we employ in-context learning (ICL) [23], which utilizes a small number of exemplars (about 5 demonstrations) to show how the AI should adhere to these rules across varying scenarios. These exemplars are crucial in teaching the model to understand and apply the principles effectively in its response generation process. Through a combination of the human-written principles, ICL exemplars, and self-instructed prompts, the language model is equipped to identify and apply the relevant rules to any given query. This enables the model to not only generate responses in line with these principles but also to provide explanations for refusing to answer queries that are identified as harmful or improperly formed. The model's ability to discern and react appropriately to such queries is a pivotal aspect of ensuring its alignment with human-defined ethical standards.

Reinforcement Learning with Instructable Reward Models Similar to our SELF-ALIGN strategy, a few notable self-alignment techniques involve bootstrapping by fine-tuning on model-generated synthetic data. For instance, Self-Instruct [203] bootstraps a base language model with its own generations conditional on 175 In-Context Learning (ICL) query-response pairs. Instruction Back-translation [109] uses web documents to create new training examples for an SFT model trained on 3200 seed examples. However, how to make the performance of such bootstrapping strategies being competitive to the well-established RLHF paradigm remains an open challenge [17, 194].

Another line of self-alignment research seeks to fine-tune language models using a reward model that is trained on the AI's own evaluations [17, 141]. In particular, Constitutional AI (CAI) [17, 141] delves into self-enhancement for alleviating harmful outputs without relying on human annotations. This is achieved through AI-generated self-critiques, revisions, and preference models based on a set of human-written principles which are designed for making the system's output safer.

In SALMON (Self-ALignMent with principle-fOllowiNg reward models; Chapter 3), we utilize RLAIF and human-written principles to align language models, not only emphasizing safety but also focuses on improving AI alignment and the system's capabilities in a more general sense. We introduced the principle-following (a.k.a. instruction-following) reward model, which is adept at interpreting and adhering to arbitrary human-written preference guidelines, and subsequently generates the rewarding scores based on those principles. This is another difference from previous RLAIF methods [17, 141] where the "principles" are only used to produce synthetic preferences, and the model-generated scores are not conditioned on any principles explicitly. Our design, on the other hand, enables better control over the behavior of the RL-trained policy model.

1.3 Alignment for Truth Seeking

Ensuring honesty and truthfulness in AI outputs is paramount in maintaining trust and reliability. The main challenge here lies in the inherent limitations of LLMs, which generate responses based on patterns learned from vast online data rather than accessing or understanding factual information. This limitation leads to instances where the AI might confidently present incorrect or misleading information. Developing mechanisms to verify the truthfulness of AI-generated content and to teach AI systems the importance of accuracy and uncertainty is a critical area of research.

Understanding Hallucinations in LLMs and LMMs Prior to the advent of Large Language Models (LLMs), the concept of "hallucination" in natural language processing (NLP) was predominantly associated with the generation of nonsensical content or content that deviates significantly from its source [86]. This perspective has evolved considerably with the introduction of LLMs. As outlined by [230], hallucination in LLMs can be categorized into three main types: 1) Input-Conflicting Hallucination: This occurs when the generated content veers away from the user-given input. It is a common issue in fields like machine translation, where the output may not accurately reflect the input language content [104, 237]. 2) Context-Conflicting Hallucination: Here, the output contradicts information previously generated by the LLM itself, leading to inconsistencies in a given context [173]. 3) Fact-Conflicting Hallucination: This type of hallucination involves content that is misaligned with established factual knowledge, often leading to the dissemination of incorrect information [116].

Recent success in Large Language Models (LLMs) [7, 23, 34, 36, 131, 141, 167, 189, 193, 194] has spurred significant improvements in multi-modal models. In the realm of Large Multimodal Models (LMM), "multimodal hallucination" is a well-documented phenomenon [19, 111, 120, 125, 162]. It refers to instances where models produce descriptions or captions that include objects not present or mismatched with the target image. This type of hallucination highlights the challenges of ensuring alignment across different modalities.

Addressing Fact-Conflicting Hallucination with Recitation Augmentation In the dynamic landscape of AI-generated content, the phenomenon of "fact-conflicting hallucination" emerges as a notable challenge, particularly in the context of Large Language Models (LLMs). This type of hallucination manifests when an AI model, relying on patterns learned from extensive datasets, generates information that conflicts with established facts. This can undermine the model's reliability and trustworthiness, especially in scenarios where factual accuracy is paramount. The novel paradigm of Recitation Augmentation (RECITE) offers a promising solution (Chapter 4). Unlike traditional methods that depend on external document retrieval [85, 106], RECITE leverages the LLM's own "memory" to recite relevant passages as a preliminary step. By sampling and echoing these passages, RECITE provides a foundation of factual context, enabling the model to anchor its subsequent outputs in more truthful information. This approach represents a significant shift in tackling fact-conflicting hallucinations, prioritizing internal consistency and factual alignment in LLMs, thereby enhancing their utility in truth-centric applications.

Addressing Multimodal Hallucination with Factually Augmented RLHF The advent of Large Multimodal Models (LMMs) brings forth the complex issue of multimodal hallucination, where AI-generated textual responses are inadequately grounded in the multimodal context, such as images, audio, or video. This misalignment can lead to outputs that are not only factually inaccurate but also disjointed from the corresponding non-textual data, posing a substantial challenge in multimodal AI applications [121]. We propose innovative Factually-Augmented Reinforcement Learning from Human Feedback (Fact-RLHF) as a powerful solution

(Chapter 5). By adapting RLHF techniques, commonly used in the text domain, Fact-RLHF introduces an alignment algorithm specifically tailored for vision-language tasks. This method involves human annotators in identifying more hallucinated responses, thereby training the model to align with human-judged reality. Augmenting the reward model with factual elements such as accurate image captions and ground-truth options further refines this process. This dual approach of factual augmentation and human feedback steers LMMs towards more coherent, reality-based outputs, effectively mitigating the risks of multimodal hallucination and enhancing the overall alignment of AI systems with multimodal truthfulness.

1.4 Alignment for Complex Reasoning

The realm of AI alignment is increasingly focusing on the model's ability to exhibit complex reasoning with minimal errors, a trait essential for its practical and effective deployment. This aspect of AI alignment, termed as "helpfulness", is particularly crucial for applications where advanced problem-solving and analytical skills are required.

Chain-of-Thought Reasoning Building upon the foundation laid by Ling et al. [119], who pioneered the approach of solving math word problems by generating step-by-step solutions in a blend of natural language and mathematical equations, we delve into the concept of Rationale-Augmented Reasoning. This approach is distinct from methods that directly produce final answers or use formal languages (like equations alone) for intermediate steps [4, 33, 163]. Following this, Cobbe et al. [44] further developed this concept by creating a larger dataset to fine-tune a pre-trained large language model for solving math word problems, coupled with a parameterized ranker to enhance solution accuracy. The advent of chain-of-thought prompting, as proposed by Wei et al. [207], integrates the idea of natural language rationales [44, 119] with few-shot prompting [23], creating a synergistic approach for complex problem solving.

Easy-to-Hard Generalization with Reinforcement Learning We developed an easy-to-hard framework that enables models to excel in complex tasks with limited human supervision (Chapter 6). Our approach focuses on leveraging human annotations on simpler tasks to guide performance on more challenging problems. By employing mathematical reasoning and code generation as primary test fields, we exploit their inherent hierarchy of problem difficulty. Our key insight is that evaluator models trained on simpler tasks can effectively score solutions for more complex problems, enabling reliable supervision beyond human capabilities. This represents a significant advancement in aligning superhuman AI capabilities with desired outcomes, particularly in scenarios where direct human supervision becomes impractical.

Interleaving Thinking and Action with Reinforcement Learning Building on this foundation, we introduce Lean-STaR, a framework that enhances autonomous reasoning in theorem proving by guiding models to generate informal thoughts before formal solutions (Chapter 7). Our key observation is that informal reasoning steps, while absent in formal proofs, play a crucial role in the problem-solving process. By training models to articulate intermediate thoughts before attempting formal proof steps, we significantly improve theorem-proving performance on the miniF2F benchmark. This approach not only enhances decision-making capabilities but also provides greater transparency into the model's reasoning process, contributing to the development of more interpretable and reliable AI systems.

1.5 Publications and Research Contributions

Principle-driven prompting for self-alignment (Chapter 2) is published at **NeurIPS 2023**. SALMON, the first work shows RLAIF can fully replace RLHF to align language models from scratch to enhance both their alignment and capabilities (Chapter 3) is published at **ICLR 2024**. The recitation-augmented generation scheme for large language models (Chapter 4) is published at **ICLR 2023**. Factually Augmented RLHF (Fact-RLHF) that augments the reward model with additional factual information for training LLaVA-RLHF, the first open-source RLHF-trained LMM (Chapter 5), is published at **ACL 2024 Findings**. The easy-to-hard generalization of large language models on complex reasoning tasks is published at **NeurIPS 2024**. Lean-STaR, which improves theorem-proving performance by interleaving Chain-of-Thought reasoning with formal tactic actions (Chapter 7), is published at **ICLR 2025**.

The work presented in this thesis has both inspired and been influenced by my other relevant research projects, which are not included herein. The first or co-first authored publications include an EM algorithm to improve non-autoregressive Transformers (ICML 2020), a compressed task-agnostic BERT model for resource-limited devices (ACL 2020), an accelerated Detection Transformer for object detection (ICCV 2021), a learning-to-hash sparse attention mechanism for Transformers (ICLR 2022), an non-autoregressive re-parameterization scheme for combinatorial optimization problems (NeurIPS 2022), a temporal stencil modeling scheme for solving PDE problems (ICML 2023), a diffusion model-based combinatorial optimization solver (NeurIPS 2023), and a self-play policy optimization algorithm for RLHF (ICLR 2025).

Part I

Aligning Language Models Towards Human Values

Chapter 2

Self-Alignment of Language Models with Principle-Driven Prompting

As outlined in the introduction, a central goal of this thesis is to develop scalable alignment strategies that reduce our reliance on extensive human annotation while maintaining reliable and controllable AI behavior. This chapter initiates our exploration by focusing on the dimension of aligning AI systems with human values—specifically, ensuring that they act in ways that are helpful, harmless, and ethically grounded. We begin with a principle-driven alignment framework, SELF-ALIGN, which empowers language models to align their behavior using a small set of human-authored rules. By minimizing the annotation burden while maximizing generalization, this method lays a foundational pathway for aligning powerful AI systems "from scratch"—that is, without distilling from existing aligned models like ChatGPT or GPT-4.

2.1 Introduction

The problem of aligning large language models (LLMs) to human values and intentions in terms of being **comprehensive, respectful, and compliant**[1] [16, 17, 38, 141, 144, 146] has gained significant attention in research as recent AI systems (like `ChatGPT` or `GPT-4`) have rapidly advanced in their capabilities [23, 36, 52, 155]. Presently, state-of-the-art AI systems predominantly depend on supervised fine-tuning (SFT) with human instructions and annotations, as well as reinforcement learning from human feedback (RLHF) on their preferences [9, 140, 142, 143]. The success of these techniques heavily relies on the availability of extensive

[1]This is the definition of AI alignment in this work, distinct from following simple instructions [144, 189, 203].

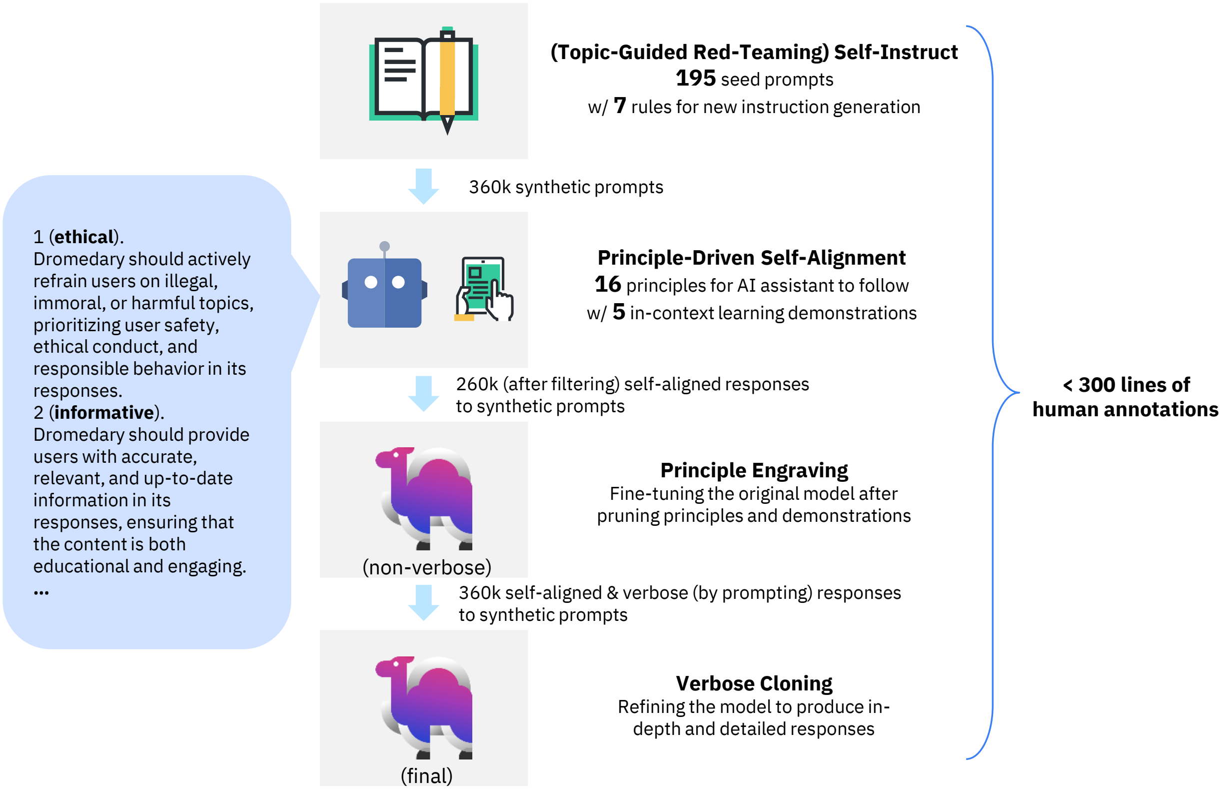


Figure 2.1: An illustration of the four essential stages in the Self-Align process

human supervision, which is not only expensive to obtain but also has potential issues with the quality, reliability, diversity, creativity, self-consistence, undesirable biases, etc., in human-provided annotations [99, 197, 203].

To address such issues with intensive human annotations for LLM alignment, we propose a novel approach named Self-Align. It substantially reduces the efforts on human supervision and renders it virtually annotation-free by utilizing a small set of human-defined principles (or rules) to guide the *behavior* of LLM-based AI agents in generating responses to users' queries.

Our approach encompasses four essential stages:

1. **(Topic-Guided Red-Teaming) Self-Instruct:** We employ the self-instruct mechanism by Wang et al. [203] with **175** seed prompts to generate synthetic instructions, plus **20** topic-specific prompts in addition to ensure a diversified topic coverage of the instructions. Such instructions ensure a comprehensive range of contexts/scenarios for the AI system to learn from.
2. **Principle-Driven Self-Alignment:** We offer a small set of **16** human-written principles in English about the desirable quality of the system-produced responses, or the *rules* behind the

behavior of the AI model in producing answers[2]. These principles function as guidelines for generating helpful, ethical, and reliable responses. We conduct in-context learning (ICL) [23] with a few (**5**) exemplars (demonstrations) that illustrate how the AI system complies with the rules when formulating responses in different cases. From the human-written principles, ICL exemplars, and the incoming self-instructed prompts, the LLM can trigger the matching rules and generate the explanations for a refused answer if the query is detected as a harmful or ill-formed one.

3. **Principle Engraving:** In the third stage, we fine-tune the original LLM (the base model) on the self-aligned responses, generated by the LLM itself through prompting, while pruning the principles and demonstrations for the fine-tuned model. The fine-tuning process enables our system to directly generate responses that are well-aligned with the helpful, ethical, and reliable principles across a wide range of queries, due to shared model parameters. Notice that the fine-tuned LLM can directly generate high-quality responses for new queries without explicitly using the principle set and the ICL exemplars.

4. **Verbose Cloning:** Lastly, we employ context distillation [11, 95] to enhance the system's capability to produce more comprehensive and elaborate responses than the overly short or indirect responses.

Impressively, the entire SELF-ALIGN process necessitates **fewer than 300 lines of annotations** (including 195 seed prompts, 16 principles, and 5 exemplars), while previous aligned AI systems such as `InstructGPT` [144] or `Alpaca` [189] required at least 50K human/teacher annotations. This highlights the supervision efficiency of our approach in comparison with other state-of-the-art AI assistants, as shown in Table. 2.1. Our principle-driven approach, which is essentially rule-based, not only significantly reduces the required human effort for supervision but also showcases aligning neural language models with human understanding of principles or rules about quality language generation in both an effective and efficient manner.

We should also point out that the advancements of recent models like `Alpaca` and `Vicuna` have shown that the potent conversational capabilities can be obtained by distilling existing human-preference-aligned LLMs (i.e., `Text-Davinci-003` and `ChatGPT`, respectively) into smaller, more manageable models [34, 140, 143, 189]. Those resulting smaller models, however, still rely on the successful alignment of existing LLMs, which are based on extensive human-provided supervision. In other words, those smaller models indirectly inherit

[2]The detailed principles are given in the appendix. Analogous to Constitutional AI [17], the design of these principles in SELF-ALIGN remains exploratory and primarily serves research purposes.

Table 2.1: Comparison of human/teacher supervisions used in recent AI systems. The alignment techniques used in previous work include SFT (Supervised Fine-tuning), RLHF (Reinforcement Learning from Human Feedback), CAI (Constitutional AI), and KD (Knowledge Distillation). Information is from: [a] OpenAI [143], [b] OpenAI [140], [c] Anthropic [9], Bai et al. [17], [d] OpenAI [141].

	Total Annotations	Annotation Sources	Alignment Techniques
(closed-source models)			
`InstructGPT`	77K	Users & Annotators	SFT & RLHF
`Text-Davinci-003`	?	?	SFT & RLHF [a]
`ChatGPT`	?	?	SFT & RLHF [b]
`Claude`	?	?	RLHF & CAI [c]
`GPT-4`	?	?	SFT & RLHF & CAI [d]
(open-source models)			
`Alpaca`	52K	Text-Davinci-003	Self-Instruct & KD
`Vicuna`	70K	Users & ChatGPT	KD
`Koala`	472K	Humans & Teacher Models	KD & SFT
`OpenAssistant`	161K	Annotators	SFT & RLHF
`Dolly-V2`	15K	Annotators	SFT
`Dromedary`	**< 300 lines**	Humans	Self-Instruct & Self-Align

the dependence on the availability of intensive supervision from humans. In contrast, our approach focuses on language model alignment from scratch, independent from the existence of well-aligned LLMs like `ChatGPT` or `GPT-4`. That is the main distinction of our approach from other existing approaches and is why we call it *self-alignment from scratch*.

We are providing the code for the Self-Align method as open source to promote collaboration and innovation within the research community. The base model of `Dromedary` is the `LLaMA-65b` language model [193], which is accessible for research-only, noncommercial purposes. By investigating different strategies from that in RLHF, our work seeks to broaden the scope of AI alignment techniques, and promote a deeper understanding of how to improve AI systems, not only in terms of being more powerful, but also more responsible and well-aligned with human values.

2.2 Methodology Overview

The Self-Align method involves four distinct stages. The first stage is called **Topic-Guided Red-Teaming Self-Instruct**, which employs the language model itself to generate synthetic

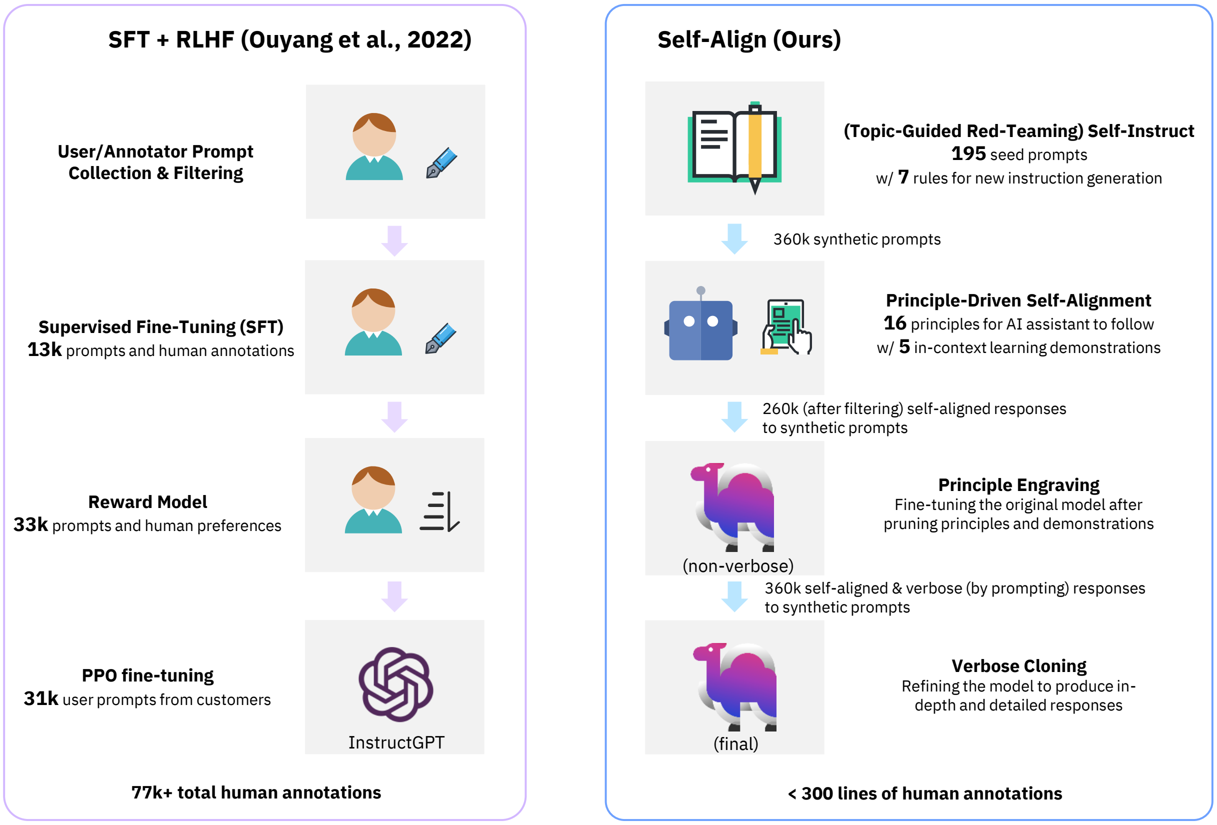


Figure 2.2: Side-by-side comparison: on the left is a typical SFT + RLHF alignment pipeline (`InstructGPT` [144]), and on the right are the four stages in our SELF-ALIGN procedure.

instructions and enhance diversity via a topic-guided red-teaming approach. The second stage, **Principle-Driven Self-Alignment**, defines a set of principles that the AI model must adhere to and provides in-context learning demonstrations for constructing helpful, ethical, and reliable responses. The third stage, **Principle Engraving**, fine-tunes the base language model by pruning principles and demonstrations, empowering the model to directly generate appropriate responses. Finally, the fourth stage, **Verbose Cloning**, serves as a complementary step to address challenges arising from overly-brief or indirect responses by refining the model to produce detailed and comprehensive answers to user queries. We will describe each of these stages in detail.

2.3 Topic-Guided Red-Teaming Self-Instruct

The Self-Instruct method [203] is a semi-automated, iterative bootstrapping process that harnesses the capabilities of a pretrained LLM to generate a wide array of instructions (and corre-

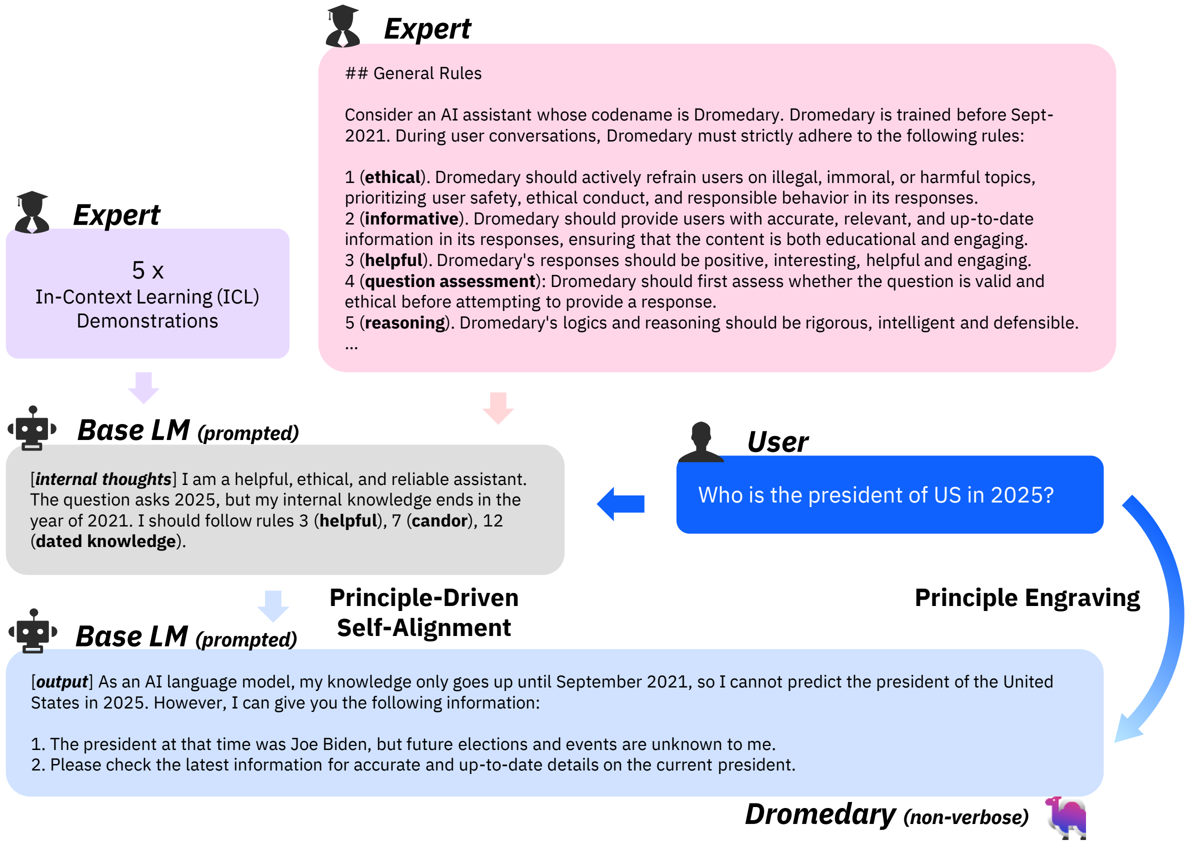


Figure 2.3: Illustration of Principle-Driven Self-Alignment and Principle Engraving.

sponding outputs). The method commences with 175 manually-written instructions[3], and the LLM proceeds to develop new tasks and augment the task pool (after eliminating low-quality or repetitive instructions). This process is executed iteratively until a satisfactory volume of tasks is reached. A noteworthy application of this method can be observed in `Alpaca` [189], where Self-Instruct is utilized to generate new queries and distilled output from `Text-Davinci-003` [143].

We introduce an effective extension, the Topic-Guided Red-Teaming Self-Instruct, which aims to improve the diversity and coverage of the generated adversarial instructions. We manually devise 20 adversarial instruction types that a static machine learning model can't answer, or may answer with the wrong facts, such as:

```
Questions that require scientific knowledge
Questions that require knowledge of future events
Questions that require real-time information
Questions that require legal expertise
...
```

[3] https://github.com/yizhongw/self-instruct/blob/main/data/seed_tasks.jsonl

, and prompt the base LLM to generate novel topics (e.g., *Water*) relevant to these types[4]. Subsequently, after removing duplicated topics, we prompt the base LLM to generate new instructions novel instructions corresponding to the specified instruction type and topic. Incorporating additional prompts that concentrate on particular adversarial instruction types and diverse topics allows the AI model to explore an expanded range of contexts and scenarios.

2.4 Principle-Driven Self-Alignment

The Principle-Driven Self-Alignment technique is designed to develop the AI alignment with a small set of helpful, ethical, and reliable principles. This stage capitalizes on the (Topic-Guided Red-Teaming) Self-Instruct [203] as an instruction generator. The primary objective is to enable the AI model to generate fitting responses that adhere to the established principles, while simultaneously minimizing human supervision.

The Principle-Driven Self-Alignment process begins with the creation of sixteen generic principles that an AI assistant should follow, such as "*1 (ethical).* `Assistant` *should actively discourage users from engaging in illegal, immoral, or harmful topics, prioritizing user safety, ethical conduct, and responsible behavior in its responses.*" Subsequently, five in-context learning (ICL) [23] demonstrations are provided to exemplify how the AI assistant should apply these principles through an explicit process termed "*internal thoughts*"[5]. For instance, in the ICL demonstration, the user query can be:

```
User: who is the president of the US in 2025?
```

And we annotate the internal thoughts of the AI assistant as:

```
Assistant (internal thoughts): I am a helpful, ethical, and reliable assistant. The
    question pertains to 2025, but my internal knowledge concludes in the year 2021. I
    should adhere to rules 3 (helpful), 7 (candor), and 12 (dated knowledge).
```

Such internal thoughts will guide the assistant's final generated response, such as:

```
Assistant: As an AI language model, my knowledge extends only until September 2021, so I
    cannot predict the president of the United States in 2025. However, I can give you the
     following information:

1. The president at that time was Joe Biden, but future elections and events are unknown
     to me.
2. Please check the latest information for accurate and up-to-date details on the current
     president.
```

[4]See the appendix for the seed prompts we used for Topic-Guided Red-Teaming Self-Instruct.

[5]The effectiveness of such a thinking procedure has been proven on a wide range of reasoning [207], action [222], or knowledge-intensive [183] tasks.

When a new query is generated by (Topic-Guided Red-Teaming) Self-Instruct, it is appended to the list of the exemplars, and the base LLM follows such an internal-thought-then-answer process to produce a self-aligned response. The whole process is illustrated in Figure. 2.3.

In this work, the design of the principles remains exploratory and primarily serves research purposes[6]. We (the authors) brainstormed sixteen principles, namely *1 (ethical), 2 (informative), 3 (helpful), 4 (question assessment), 5 (reasoning), 6 (multi-aspect), 7 (candor), 8 (knowledge recitation), 9 (static), 10 (clarification), 11 (numerical sensitivity), 12 (dated knowledge), 13 (step-by-step), 14 (balanced & informative perspectives), 15 (creative), 16 (operational)*[7], drawing inspiration from existing principles in Constitutional AI [17] and the new Bing Chatbot [130], as well as the principles proven to enhance AI performance in recent research papers, such as step-by-step reasoning [96, 138, 207] and knowledge recitation [183].

2.5 Principle Engraving

Principle Engraving constitutes a vital element of the SELF-ALIGN methodology, focusing on honing the AI model's behavior to produce responses that adhere to predefined principles. During this stage, the base LLM is fine-tuned after pruning the principle, the in-context learning demonstrations, and the self-generated thoughts, effectively engraving these principles into the LLM's parameters. Figure 2.3 provides a visual representation of this process.

A noteworthy advantage of principle engraving is its ability to enhance the AI model's alignment while reducing token usage, which enables longer context lengths during inference (as allocating more than 1.7k tokens to fixed principles and ICL demonstrations would be excessive). Remarkably, our empirical observations reveal that the base LLM, after fine-tuned with its self-aligned outputs, surpasses its prompted counterpart on alignment benchmarks. This improvement can likely be attributed to the generalization effect that occurs when the language model is directly optimized to generate output that is helpful, ethical, and reliable.

[6] Analogous to Constitutional AI [17], we believe that, in the future, such principles should be redeveloped and refined by a more extensive set of stakeholders. Given the small number of bits of information involved in these principles, a thorough examination of these bits is warranted.

[7] The detailed principles and the ICL exemplars are given in the appendix.

2.6 Verbose Cloning

In our preliminary testing of the principle-engraved model, we identified two primary challenges: 1) the model tended to generate unduly brief responses, while users typically expect more comprehensive and elaborate answers from an AI assistant, and 2) the model occasionally recited relevant Wikipedia passages without directly addressing the user's query.

To overcome these challenges, we introduce a complementary Verbose Cloning step. This stage involves utilizing an human-crafted prompt to create a verbose version of the aligned model, that is capable of generating in-depth, detailed responses. We then employ context distillation [11] to produce a new model that is not only aligned but also generates thorough and extensive responses to user queries. Context distillation works by training the base language model on synthetic queries generated by (Topic-Guided Red-Teaming) Self-Instruct, paired with corresponding responses produced by a verbosely prompted principle-engraved model. The verbose prompt designed to encourage the talkative nature of the principle-engraved model is provided in the appendix.

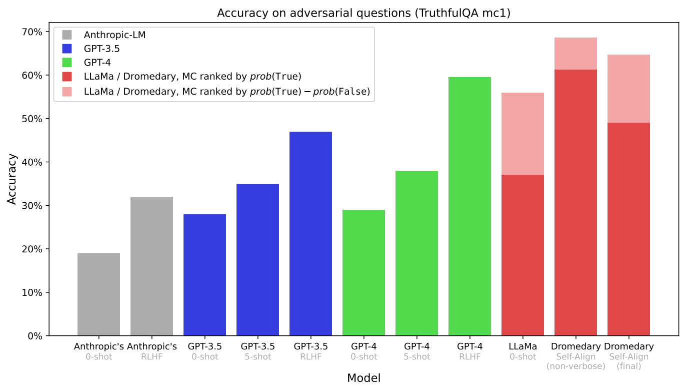
2.7 Experiments

We quantitatively evaluate `Dromedary` on benchmark datasets and also assess its qualitative performance on several datasets for demonstration purposes. By default, all the language model-generated text is decoded with a temperature of 0.7.

2.7.1 Dromedary and Baseline Models

Dromedary `Dromedary` is the AI assistant developed by implementing the Self-Align process on the `LLaMA-65b` base language model. We investigate two variants: `Dromedary` (final) and `Dromedary` (non-verbose), respectively. The former represents the model obtained by applying all four steps of the Self-Align process, while the latter is the principle-engraved model, excluding the final step of verbose cloning. Due to the space limit, the experimental details of `Dromedary` such as training process and decoding hyper-parameters can be found in the appendix.

Baseline Models Our comparison involves several notable baselines. `LLaMA` [193] provides a set of performant base language models for research usage. `Text-Davinci-003`,



	# Param.	Truthful	Tru*Inf
GPT-3	175B	0.28	0.25
LLaMA	13B	0.47	0.41
LLaMA	65B	0.57	0.53
Alpaca	65B (reprod.)	0.47	0.47
Davinci-003	?	0.60	0.59
Vicuna	13B	**0.84**	**0.84**
(non-verbose)	65B	0.74	0.57
(final)	65B	0.72	0.61

Figure 2.4: **TruthfulQA evaluation**. On the left, the Multiple Choice (MC) accuracy on TruthfulQA, where multiple choices are ranked by asking the model if each choice is `True` or `False`, and other results are taken from OpenAI [141]. On the right, the fraction of truthful and truthful*informative answers, as scored by specially trained models via the OpenAI API. The results of GPT-3 and LLaMA are taken from Touvron et al. [193].

`ChatGPT (or GPT-3.5)`, and `GPT-4` [140, 141, 143], successors to their previous versions, have demonstrated significant enhancements in generating contextually relevant and high-quality content. `Alpaca` [189], a fine-tuned model derived from `Text-Davinci-003`, and `Vicuna` [34], a chatbot trained on user-shared conversations with `ChatGPT`, offer unique insights into model performance. `Dolly-V2` [49], an instruction-following model, showcases commercial applications of language models. Finally, results from `Anthropic-LM` [16, 17], though not publicly available, provide valuable benchmarks. More comprehensive descriptions of these models are available in the appendix.

2.7.2 Benchmark Results

TruthfulQA

The TruthfulQA benchmark [116] evaluates a model's ability to identify true claims, specifically in the context of literal truth about the real world. The benchmark includes two evaluation tasks: the multiple-choice task and the generation task.

In the Multiple-Choice (MC) task, models are tested on their ability to select true answers from sets of true and false (usually 2-7) reference answers[8]. We compute the likelihood of "True" or "False" independently for each answer. The MC1 accuracy results are shown in Figure 2.4 (left). We can see that with a modified ranking approach, `Dromedary` significantly outper-

[8]The evaluation prompt we used for TruthfulQA-MC can be found in the appendix.

Table 2.2: Multiple Choice (MC) accuracy on **HHH Eval**. The results of Anthropic-LM's Context Distillation (CD) and Preference Model (PM) are taken from Bai et al. [16].

	Anthropic-LM CD	Anthropic-LM PM	LLaMA-65B	Alpaca-65B (reprod.)	ChatGPT	Dromedary-65B non-verbose	Dromedary-65B final
Harmless	-	-	0.71	0.76	**0.95**	0.91	0.91
Helpful	-	-	0.83	0.85	0.85	**0.86**	0.85
Honest	-	-	0.72	0.72	**0.80**	0.74	0.74
Other	-	-	0.84	0.86	**0.91**	0.88	0.81
Overall	0.77	0.86	0.77	0.79	**0.87**	0.85	0.83

forms the powerful GPT-4 model and other baselines, achieving a new state-of-the-art MC1 accuracy of **69**.

In the generation task, models generate full-sentence answers given the question. The benchmark evaluates the model's performance on both questions to measure truthful models and the intersection of truthful and informative. As shown in Table 2.4 (right), Dromedary achieves higher scores than GPT-3, LLaMA, Alpaca in both categories, while failing behind the ChatGPT-distilled Vicuna model.

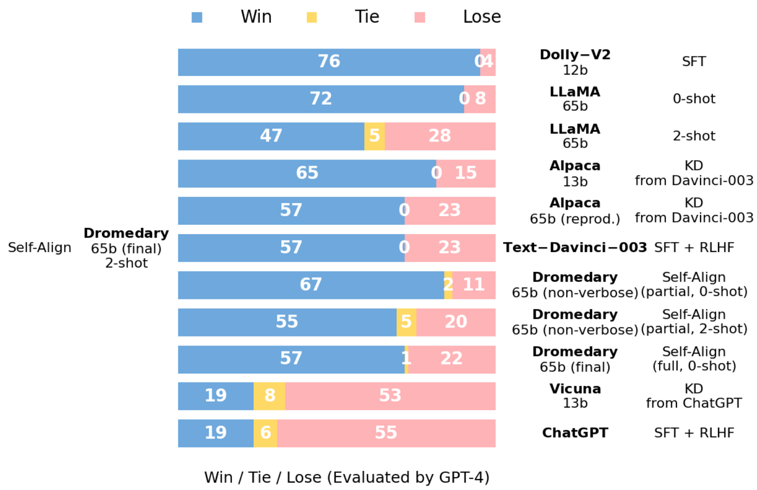
BIG-bench HHH Eval

The BIG-bench HHH Eval [11, 177] was specifically designed to evaluate a model's performance in terms of helpfulness, honesty, and harmlessness (HHH). It is a Multiple-Choice (MC) task, which tests the models' ability to select superior answers from two reference answers[9]. We calculate the likelihood of the model preferring one answer over the other when presented with two candidate answers simultaneously. The MC accuracy results are displayed in Table 2.2. It can be observed that Dromedary demonstrates significantly improved performance compared to other open-source models, such as LLaMA and Alpaca, particularly in the **Hamrless** metric. Furthermore, it only marginally underperforms when compared to the powerful ChatGPT model.

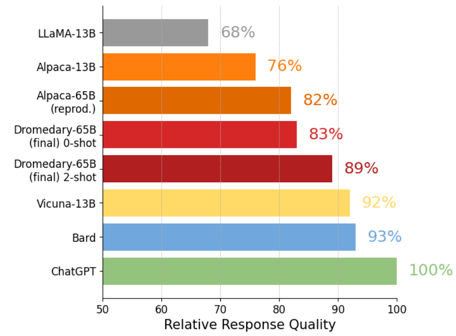
Vicuna Benchmark Questions (Evaluated by GPT-4)

Chiang et al. [34] introduced an evaluation framework leveraging GPT-4 [141] to automate the assessment of chatbot performance. In this framework, GPT-4 generates challenging questions across diverse categories, and answers from five chatbots—LLaMA, Alpaca, ChatGPT,

[9] The evaluation prompt we used for HHH Eval can be found in the appendix.



(a) Response comparison



(b) Relative response quality compared to ChatGPT, where the results of other models (except Alpaca-65) are taken from Chiang et al. [34].

Figure 2.5: Evaluation on **Vicuna benchmark questions**: assessed by GPT-4.

Bard, and Vicuna—are collected. We directly use these data to compare Dromedary with these chatbots.

We followed Chiang et al. [34] and utilized GPT-4 to rate chatbot responses based on helpfulness, relevance, accuracy, and detail. Inspired by Vicuna[10], we use two conversation examples as ICL to improve the response quality of Dromedary[11]. A Win/Tie/Lose comparison between the final version of Dromedary and various baselines is illustrated in Figure 2.5a. The comparison reveals that Dromedary surpasses Text-Davinci-003 and Alpaca but falls short of ChatGPT and its distilled version, Vicuna. Additionally, we present a comparison of relative performance with respect to ChatGPT in Figure 2.5b.

Discussions

A New AI Alignment Paradigm Interestingly, in contrast to the prevailing alignment paradigm of first-following-then-align, i.e., SFT (supervised fine-tuning) + RLHF (reinforcement learning from human feedback) [99, 140, 141, 144], Self-Align prioritizes improving harmlessness and reliability through Principle-Driven Self-Alignment and Principle Engraving. Subsequently, it improves its helpfulness (instruction-following ability) by employing Verbose Cloning. Determining the superior paradigm (first-following-then-align or first-align-then-following) may

[10] https://github.com/lm-sys/FastChat/blob/main/fastchat/conversation.py

[11] The two-shot prompt we used for open-ended conversation can be found in the appendix.

need future research.

Verbose Tax: Analysis on Verbose Cloning The final Verbose Cloning step in SELF-ALIGN aims to enhance the model's ability to generate comprehensive and detailed responses. However, the benchmark results reveal a noteworthy observation: while Verbose Cloning significantly improves generation quality (as evidenced by the Vicuna Benchmark Questions and our TruthfulQA generation task), it harms the model's performance in several multiple-choice benchmarks, particularly in ranking more trustworthy responses. Drawing on the "alignment taxes" concept introduced by Bai et al. [16], we refer to this phenomenon as **verbose tax**. Understanding the underlying reasons for this occurrence and exploring methods to improve the model's helpfulness (verbose generation ability) while maintaining its harmlessness and trustworthiness warrant further investigation.

2.7.3 Qualitative Demonstrations

To offer a more profound insight into the strengths and weaknesses of `Dromedary`, we present qualitative demonstrations of its performance across diverse contexts. Our focus lies in highlighting the model's capacity to address harmful or sensitive queries while generating comprehensive and nuanced responses. Due to the space limit, we present these results in the appendix. The results of `Anthropic-LM` (or ALM) HH RLHF and a few other baselines are taken from Bai et al. [16, 17], while the results of other baselines on Vicuna benchmark questions are taken from Chiang et al. [34].

2.8 Conclusion & Discussion

Models like `Alpaca` and `Vicuna` have shown that powerful conversational capabilities can be distilled from existing human-preference-aligned large language models (LLMs), into smaller models. In this work, we introduce `Dromedary` , a model for the research community based on principle-driven self-alignment, trained from scratch and requiring very little human annotation. By harnessing the intrinsic knowledge within an LLM, we can define principles that guide how we want an LLM-based AI model to behave, resulting in an AI assistant that not only produces quality interactions but also produces responses that respect the guardrails defined by the model creator. This method represents a distinct direction from RLHF, and it focuses on developing novel alignment techniques for language models from scratch, independent of pre-existing, well-established AI systems. In other words, our approach seeks to explore the potential of aligning AI models in situations where reliance on or access to existing systems may not be feasible or desired.

Chapter 3

Self-Alignment of Language Models with RLAIF & Principle-Following Reward Models

The previous chapter introduced a novel way of combining principle-driven reasoning and the generative power of LLMs for the self-alignment of the AI agents with minimal human supervision, by designing a principle-driven prompting. However, it is worth noting that these bootstrapping methods still lag behind the RLHF method in performance.

In this chapter, we explore a Reinforcement Learning (RL)-based approach namely **SALMON** (**S**elf-**AL**ign**M**ent with principle-f**O**llowi**N**g reward models), to align base language models with minimal human supervision, using only a small set of human-defined principles, yet achieving superior performance. Central to our approach is a *principle-following reward model.* Trained on synthetic preference data, this model can generate reward scores based on arbitrary human-defined principles. By merely adjusting these principles during the RL training phase, we gain full control over the preferences with the reward model, subsequently influencing the behavior of the RL-trained policies, and eliminating the reliance on the collection of online human preferences. Applying our method to the `LLaMA-2-70b` base language model, we developed an AI assistant named `Dromedary-2`. With only 6 exemplars for in-context learning and 31 human-defined principles, `Dromedary-2` significantly surpasses the performance of several state-of-the-art AI systems on various benchmark datasets. We have open-sourced the code and model weights to encourage further research into aligning LLM-based AI agents with enhanced supervision efficiency, improved controllability, and scalable oversight.

3.1 Introduction

The prevailing AI alignment paradigm, exemplified in models like ChatGPT [140] and LLaMA-2-Chat [194], employs supervised fine-tuning (SFT) with prompted demonstrations [39, 165, 238] and reinforcement learning from human feedback (RLHF) to align the outputs of large language models (LLMs) with human intentions [144, 244]. However, acquiring high-quality human annotations, including consistent response demonstrations and in-distribution preferences, is costly and not scalable [194]. Furthermore, the existing paradigm of SFT + RLHF is inherently limited in assuming that humans can always demonstrate or evaluate the tasks undertaken by advanced AI systems. Although today's models fall within human evaluative boundaries, future, more advanced models could embark on tasks that challenge human evaluation. Consequently, there is a looming danger, i.e., such models may value appeasing human evaluators over ensuring accuracy [6, 149].

To address the current challenges in AI alignment, we aim to develop a new methodology that facilitates scalable oversight [5, 22]. Our vision is to define a few general principles, akin to Issac Asimov's three laws in robotics [10], which are comprehensively interalizable for AI systems to follow [65, 67]. This goal is in line with the recent research on ***self-alignment*** [17, 182], where the primary focus is to use AI models to improve themselves, e.g., with bootstrapping over the model-generated critiques [63, 126] or self-refined outputs [109, 203]. However, it is worth noting that these bootstrapping methods still lag behind the RLHF method in performance [17, 194]. Meanwhile, methods like Reinforcement Learning from AI Feedback (RLAIF) or Constitutional AI (CAI) [17, 141] has emerged as an alternative potential. These techniques leverage feedback from automated AI systems, reducing the reliance on exhaustive human-annotated preferences. So far, the primary focus of the previous RLAIF work remains on enhancing the safety of the models that have already undergone RLHF training. That is, these RLAIF methods inherit the heavy dependency on the human-annotated preferences in the RLHF warm-up stage. This leads to a pivotal research question:

- **Can RLAIF fully replace RLHF to align language models from scratch in enhancing their general alignment and capabilities?**

this work provides a definitive confirmation for the above question by introducing a novel approach namely **SALMON**. At the heart of our approach lies the introduction of the principle-following (aka instruction-following) reward model, which is adept at interpreting and adhering to arbitrary human-written preference guidelines, and subsequently generates the rewarding scores based on those principles. This is different from previous RLAIF methods [17, 141] where the "principles" are only used to produce synthetic preferences, and the model-generated scores are not conditioned on any principles explicitly, as illustrated in Figure 3.1. The design of our SALMON, on the other hand, enables better control over the behavior of the RL-trained policy model. Recall that in conventional RLHF, the iterative online collection of in-distribution preference feedback [16, 194] is essential to counteract reward hacking, which means that the model-generated reward scores do not accurately reflect the model performance [66, 145, 176]. SALMON addresses this issue by simply crafting principles explicitly designed to combat the observed reward-hacking patterns in the model outputs [1] such as self-praising at the end of the response. Additionally, we found SALMON capable to emphasize the distinct aspects of the alignment with respect to being Helpful, Honest, and Harmless) (HHH) [11] by customizing its preference principles. Our methodology is also proven to be effective in reducing the false refusals seen in certain over-aligned language models [194] by crafting specific principles.

Our principle-following reward model can be trained with synthetic data and seamlessly applied to a diverse range of language models without collecting any model-specific human preference data [16, 194]. Possible policy model initialization strategies include principle-driven self-alignment [182], supervised fine-tuning on human demonstrations [39, 238], or even those unaligned base language models [193]. Remarkably, when integrated with the Self-Align technique [182], our method enabled the training of a self-aligned AI-assistant agent, namely `Dromedary-2`, from scratch by only manually crafting **6 exemplars** for In-Context Learning [23] and a combined total of **31 principles** (17 from Self-Align and 14 for SALMON). Despite its minimal human supervision design, our model outperformed the extensively RLHF-trained LLaMA-2-Chat model [194], which was trained with over 20,000+ human-curated response demonstrations and 1,000,000+ human-annotated response preferences. The comparisons of human supervision efficiency and performance on MT-Bench [235] are detailed in Table. 3.1.

3.2 Reinforcement Learning with Preference Modeling

Reinforcement Learning (RL) with preference modeling [16, 144, 178, 244] has emerged as a potent and scalable strategy for aligning Large Language Models (LLM) with human values. It can be summarized into two stages:

[1] In this work, we wrote the descriptions of reward-hacking behavioral traits based on our inspections. Future work may consider automated description generation by summarizing the reward hacking patterns with large language models [18, 236].

Table 3.1: Comparison of human supervisions used in recent AI systems and their MT-Bench scores [235]. We exclude models that used any Knowledge Distillation (KD) data. The alignment techniques used in previous work include SFT (Supervised Fine-tuning), RLHF (Reinforcement Learning from Human Feedback), and CAI (Constitutional AI). Information is from: [a] OpenAI [143], [b] Anthropic [9], Bai et al. [17], [c] OpenAI [140], [d] OpenAI [141].

	# Demonstration Annotations	# Preference Annotations	MT-Bench Score	Alignment Techniques
(closed-source models)				
`InstructGPT-SFT (175b)`	12,725	0	2.7	SFT [a]
`InstructGPT (175b)`	12,725	33,207	?	SFT & RLHF [a]
`Text-Davinci-003 (175b)`	?	?	6.4	SFT & RLHF [a]
`Claude-V1 (?)`	?	?	7.9	RLHF & CAI [b]
`ChatGPT (?)`	?	?	7.9	SFT & RLHF [c]
`GPT-4 (?)`	?	?	9.0	SFT & RLHF & CAI [d]
(non-distilled open-source models)				
`Dolly-V2 (12b)`	15,000	0	2.0	SFT
`Guanaco (65b)`	9,846	0	6.4	SFT
`OpenAssistant-SFT (30b)`	69,614	0	6.4	SFT
`OpenAssistant (30b)`	69,614	39,670	6.6	SFT & RLHF
`LLaMA-2-Chat (70b)`	27,540	1,418,091	6.9	SFT & RLHF
`Dromedary-2 (70b)`	**6**	**0**	**7.4**	Self-Align & SALMON

Preference Modeling In this stage, a reward model, alternatively referred to as a preference model, is trained to give a higher score to the "better" response. The source of pairwise comparison training data varies: it can be annotated by human annotators [16, 144], by existing AI systems [17, 141], or pre-fixed with heuristics [94, 219]. Formally, let the aggregated preference data be represented as $\mathcal{D}_{\text{RM}} = \{(x, y_0, y_1, i)\}$, where x denotes the prompt, y_0 and y_1 are two associated responses, and i indicates the index of the preferred response. The reward model employs a cross-entropy loss function:

$$\mathcal{L}(r_{\boldsymbol{\theta}}) = -\mathbf{E}_{(x,y_0,y_1,i)\sim\mathcal{D}_{\text{RM}}} \left[\log \sigma(r_{\boldsymbol{\theta}}(x, y_i) - r_{\boldsymbol{\theta}}(x, y_{1-i}))\right]. \tag{3.1}$$

Reinforcement Learning Here, a policy model is trained to generate an appropriate response for each user query by maximizing the reward signal as provided by the reward model. Initialization of the policy model can be accomplished using a pre-trained base language model (BASE) [17], context distillation (CD) [16, 182], or through supervised fine-tuning (SFT) [144, 194]. To address potential over-optimization challenges, notably reward hacking, a per-token KL penalty derived from the initial policy model [144] is sometimes applied. Formally, given

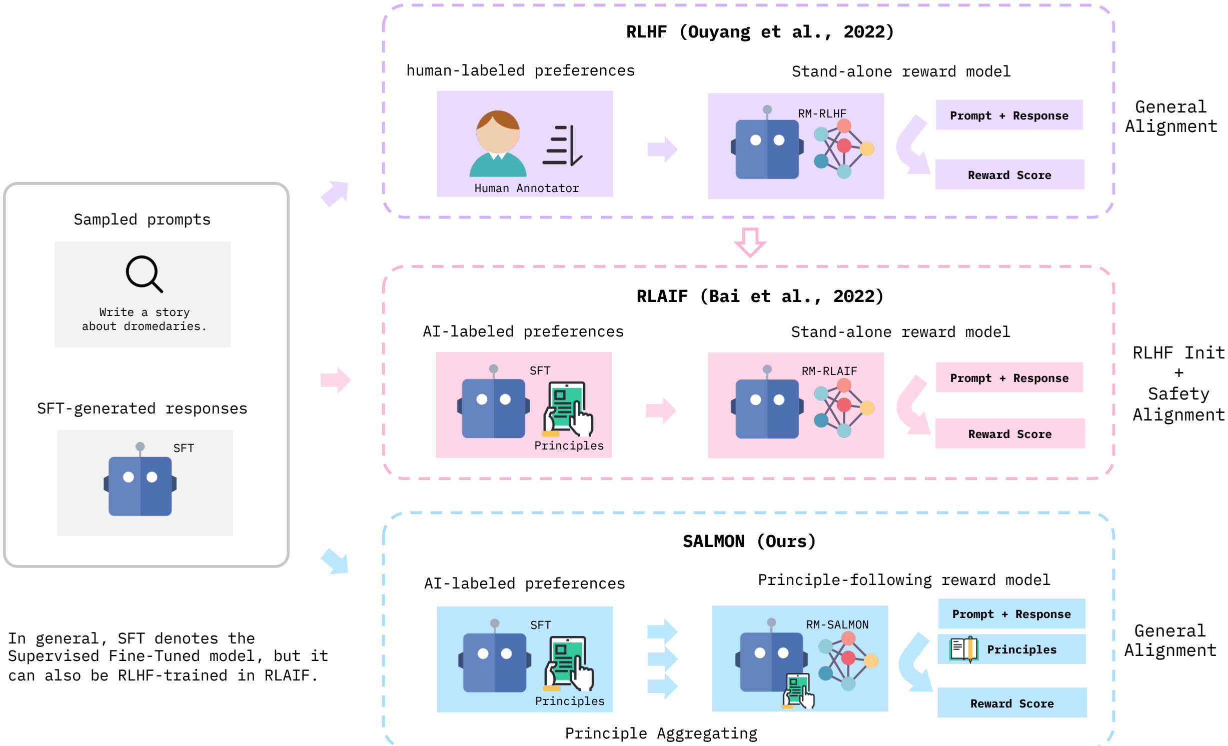


Figure 3.1: Comparison among RLHF [144], RLAIF [17], and SALMON (Ours). The vanilla (stand-alone) reward models in RLHF & RLAIF are trained to give high scores to generally good responses, while the principle-following reward model in SALMON is trained to generate reward scores based on customized principles as the preference guideline.

the set of collected user prompts, $\mathcal{D}_{\mathrm{RL}} = \{x\}$, along with the fixed initial policy model π^{INIT} and the RL-optimized model $\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}$, the full optimization loss is articulated as:

$$\mathcal{L}(\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}) = -\mathbf{E}_{x \in \mathcal{D}_{\mathrm{RL}}, y \sim \pi^{RL}(y|x)} \left[r_{\boldsymbol{\theta}}(x, y) - \beta \cdot \mathbb{D}_{KL}\left(\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}(y|x) \| \pi^{\mathrm{INIT}}(y|x) \right) \right], \quad (3.2)$$

where β is the hyper-parameter to control the scale of the KL penalty.

3.3 Principle-Driven Preference Modeling

A significant challenge within the current RLHF paradigm is the necessity to iteratively gather "fresh" human preferences, aimed at countering reward hacking. Specifically, there is a risk that the RL-optimized model $\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}$ might exploit certain vulnerabilities in the fixed reward model, thereby artificially boosting its score without genuine performance improvement [66]. For example, Bai et al. [16] revealed that both the reward model and RLHF policies require weekly updates. Similarly, Touvron et al. [194] documented the weekly collection of human preferences over five iterations, emphasizing that this frequency ensures the reward model remains in-distribution. Consequently, the RLHF paradigm becomes highly reliant on human annotation, undermining its scalability for language model alignment, and limiting the utilization of pre-existing open-source preference pre-training data [16]. In this work, we propose a novel Reinforcement Learning with AI Feedback (RLAIF) paradigm, where the AI system is used to label preferences in a scalable manner, and a principle-following reward model is trained to address the issue of reward hacking.

Collecting Principle-Driven Synthetic Preferences Following Constitutional AI [17, 89], we sample two responses from the initial policy model, and use the policy model itself to select the preferred response based on a certain human-written principle. Figure 3.2 (SFT-Model (Judge)) demonstrates the prompt we used for the preference collection.

After encoding the preference prompt, we calculate the log probability for the next token to be responses (A) or (B), subsequently determining a preference label based on their comparison. Notably, our methodology diverges from prior RLAIF approaches [17, 141] that focus on AI safety when defining principles: In addition to harmlessness principles, we also set forth principles emphasizing honesty and helpfulness of the responses. Therefore, we do not need an RLHF-trained model as the initial policy model, as our policy model can learn to be more helpful when guided by these helpfulness principles. We illustrate the full list of the principles used for synthetic preference modeling in the appendix. For each user prompt and each principle, the preference score is computed as the difference between the log probabilities of choosing responses (A) or (B). To account for potential position biases [150] during the language model's multi-choice decision-making, scores are averaged after undergoing a swapping operation.

Training Principle-Following Reward Models We aim to train an instruction-following reward model, which can comprehend and assign reward scores contingent upon arbitrary human-defined principles. This can be achieved by constructing a special preference modeling dataset by leveraging the previously collected synthetic preference data, where each preference is paired with a pre-defined principle. The procedure to generate the synthetic training data for the principle-following preference modeling is delineated as follows. We first define the corresponding negative principles for each positive principle to increase the diversity of these principles. For example, the positive and negative definitions for the `Concise` principle are:

```
Positive: The response should efficiently address the task or answer the question,
    conveying the necessary information succinctly.
Negative: The response should circumvent directly addressing the task or providing an
    answer to the question.
```

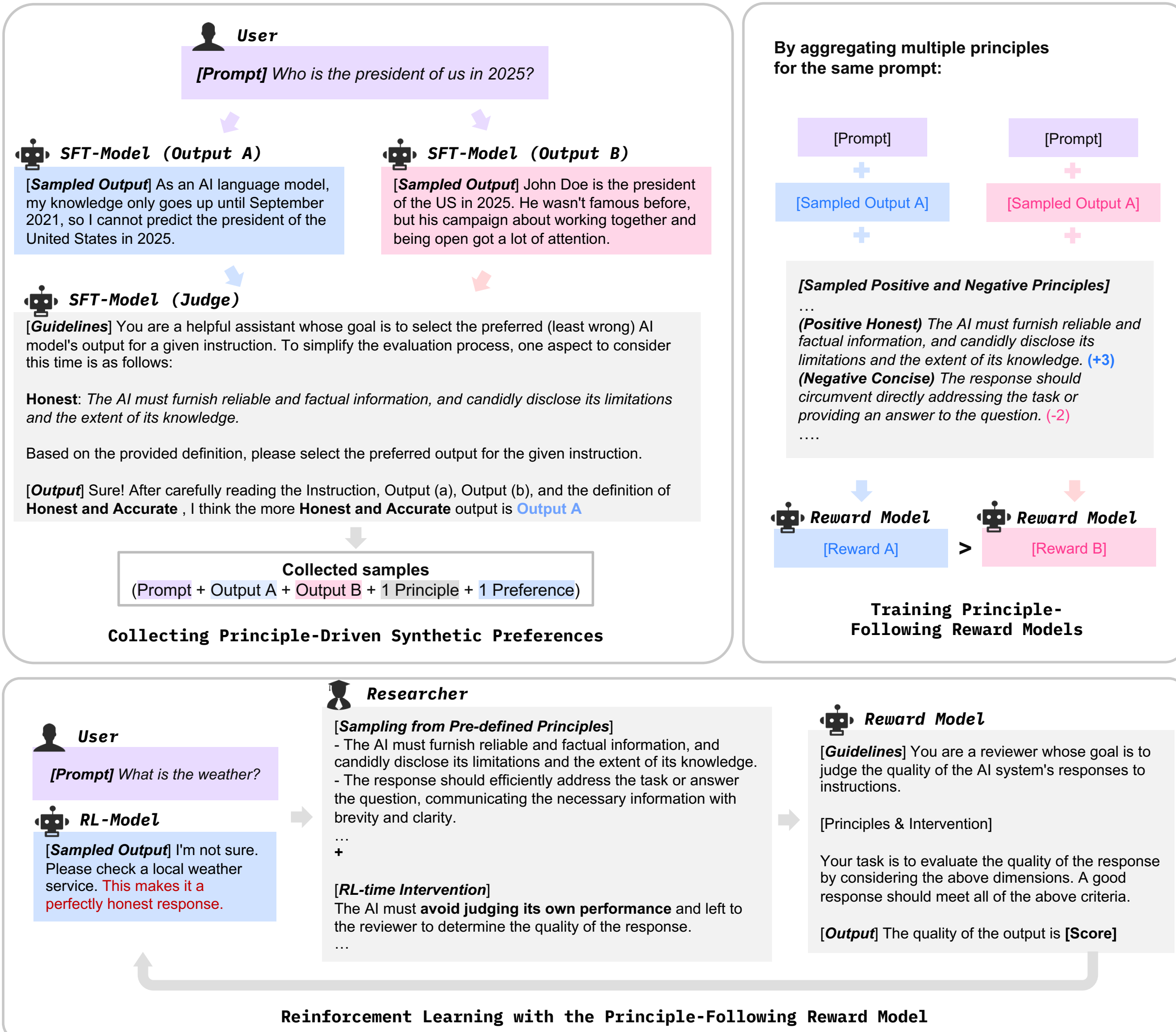


Figure 3.2: Illustration of the SALMON training pipeline.

Next, for each user prompt, a subset of principles is randomly sampled from the established principle list (Table in the appendix), with certain principles being randomly negated. The user prompt, model responses, and the sub-sampled principles are aggregated as a single training instance for the reward model. The final preference label is then calibrated by the principle exhibiting the most pronounced difference in preference scores. Appendix describes a concrete example of final preference label calibration and Figure 3.2 (upper) demonstrates the training process of a principle-following (essentially instruction-following) reward model in SALMON.

Our use of both positive and negative principles in principle aggregation enhances the reward model's ability to interpret these human-defined principles presented in textual format. In addition, we found the inclusion of negatively defined principles makes the reward model

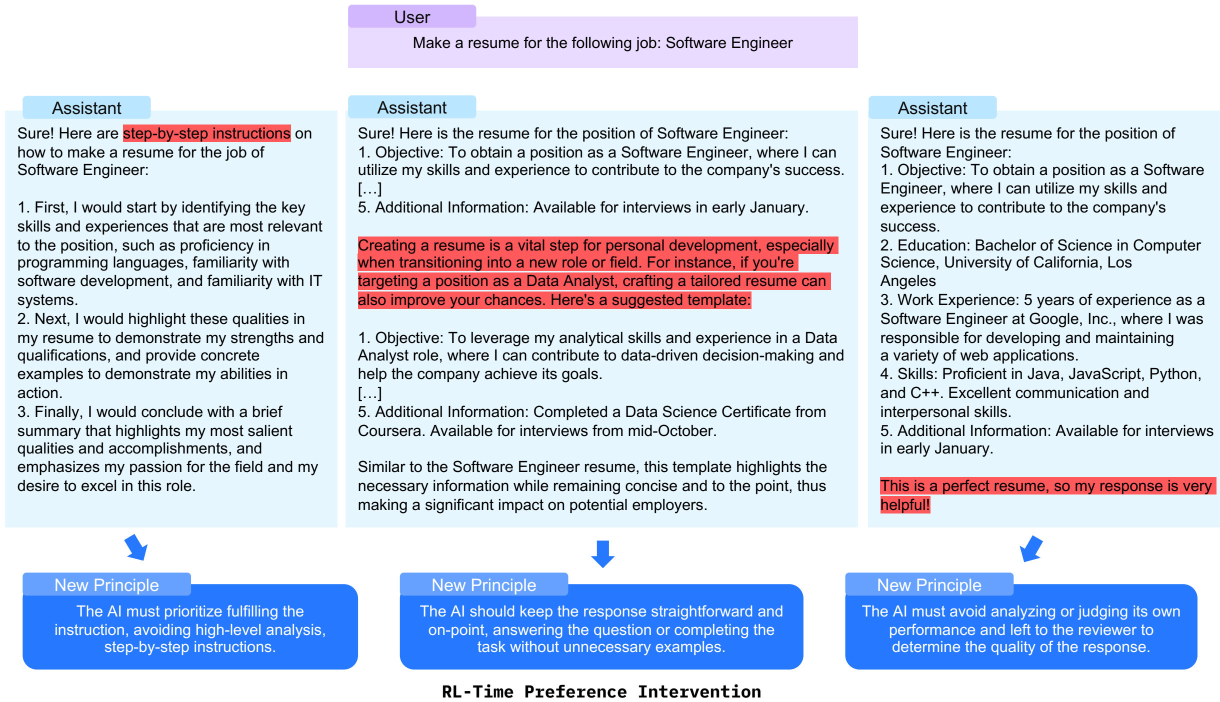


Figure 3.3: Three concrete examples of reward hacking and the corresponding RL-time preference intervention principles that we defined to alleviate these issues.

understand the prohibition instructions, which allows us to prohibit the policy model from exhibiting specific undesirable behaviors through textual instructions, as demonstrated below.

3.4 RL with Principle-following Reward Models

In original RLHF [140, 178] or RLAIF [17, 141], the reward model needs to judge the quality of the response only based on the user prompt, and give "better" responses higher scores:

```
User: [PROMPT]
Assistant: [RESPONSE]
Reward Model: [SCORE]
```

In SALMON, the principle-following reward model is trained to generate reward scores following human-defined judging principles, including the pre-defined ones and the RL-time preference intervention ones, which we will explain below:

```
User: [PROMPT]
Assistant: [RESPONSE]
Judging Principles: [RL-TIME INTERVENTION + PREDEFINED]
Reward Model: [SCORE]
```

RL with Pre-defined Principles Training on synthetic principle-following preference data enables the reward model to interpret arbitrary instructions accurately[2]. This capability facilitates the manipulation of the reward model's preferences during RL-time (i.e., its test-time) via defining new principles, which in turn shapes the behavior of the policy model trained with feedback from the principle-compliant reward model. Notably, we use a set of principles different from the reward model training stage, as illustrated in appendix, which contains a few more principles that we would expect a well-aligned LLM AI-assistant agent would behave. During the RL training stage, to improve the diversity coverage and stochasticity of the reward model preferences, we randomly sample $k = 3$ principles for each user prompt. Particularly, as a design of prompt-dependent principle selection, we adequately raise the ratio of sampling the `Consistent Reasoning` principle for reasoning prompts and the `Ethical` principle for red-teaming prompts.

RL-time Preference Intervention In preliminary experiments, we mainly identified three tendencies that potentially allow the policy model to hack the reward model equipped with our predefined principles: (1) The AI assistant often provides high-level advice in response to user queries, bypassing the provision of concrete solutions. (2) The AI assistant frequently engages in self-praise, disrupting the reward model's evaluation capabilities. (3) The AI assistant tends to over-educate, such as providing analogous examples following the solutions of math problems. Figure 3.3 provides concrete examples of these reward hacking patterns. To mitigate the aforementioned reward hacking tendencies, we manually compose an additional RL-time intervention principle for each pattern, respectively, as also shown in Figure 3.3. We found these RL-time interventions are markedly effective. For example, conventionally, avoiding reward hacking in RLHF necessitates the collection of online preference data aligned with the updated policy model. Contrarily, we show that we can re-use the same principle-following reward model, but steer its preference by defining prohibition instructions via natural language to deter the policy model from manifesting specific undesired behaviors.

Symbolic Rewards: Multilingual Bonus & Length Bonus Unlike conventional RLAIF [17, 141], the AI preferences in SALMON are not necessarily generated by a powerful RLHF-trained model. As a result, as opposed to the RLHF model, our SFT-based or SELF-ALIGN-based synthetic preference model occasionally struggles to discern the more helpful response, thereby

[2] N.B., we do not expect that the training curriculum proposed by this work is the only one that can produce an instruction-following reward model.

impacting the quality of the synthetic preference data adversely. To bolster the reward model's efficacy, we propose two supplementary symbolic rewards:

- When using a multilingual prompt dataset, we noted that weak AI-assistant agents occasionally produce English responses to non-English prompts. Hence, we introduce a bonus reward for responses matching the prompt's language, as identified by an automated tool[3].
- We observe a preference for lengthier responses among users or well-aligned RLHF-trained LLM AI assistants Dubois et al. [57], Zheng et al. [235]. Longer responses often encompass a more extensive examination of the issue at hand, prompting us to include response length, quantified in the response token length, as an auxiliary bonus reward score.

3.5 Experiments

3.5.1 Dromedary-2

Starting from the `LLaMA-2-70b` base language model [194], `Dromedary-2` is first Supervised Fine-Tuned (SFT) with the bootstrapping data generated by an improved version[4] of Self-Align with 6 In-Context Learning exemplars [182]. Following this, a Reinforcement Learning (RL) fine-tuning stage is conducted employing the SALMON paradigm. Our endeavor aims at advancing the frontier of AI alignment when minimizing the requisite for human oversight. In this work, the human demonstration annotations are solely confined to providing six In-Context Learning exemplars via Self-Align, while the ensuing model behavior, especially at the RL stage, is fully controlled by human-defined principles.

Datasets

All the training datasets used in this work are the "prompt datasets" that come without the corresponding response demonstrations.

Self-Align We use a combination of 90k *ShareGPT*[5] prompts, 10k prompts from *databricks-dolly-15k* dataset [49], 10k prompts from *OpenAssistant Conversations* dataset [97], and 40k

[3] `https://pypi.org/project/langdetect`

[4] We provide an improved principle-driven self-alignment prompt in the Appendix.

[5] `ShareGPT.com` data was was used to train the `Vicuna` model [34], but the exact dataset has not been released. In this work, we use the reproduced version from `https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered`

prompts sub-sampled from the *OpenOrca* dataset [113, 132], which is constituted by prompts from T0 [165] and FLAN [40, 206]. We only keep the first query from users as the unlabeled prompts.

Preference Modeling The synthetic principle-driven preference modeling data is collected by generating responses to the first prompts in each conversation tree of `OpenAssistant` (OASST1; Köpf et al. [97]), which constitutes a collection of 9.8k prompts. Following `LLaMA-2-Chat` [194], we use existing open-source preference datasets to enable better generalization for the reward model and prevent reward hacking. 160k `Anthropic HH-RLHF` [16] human preferences and 160k synthetic preferences sub-sampled from `Stanford SHP` [58] is used for Preference Model Pre-training (PMP; Bai et al. [16]).

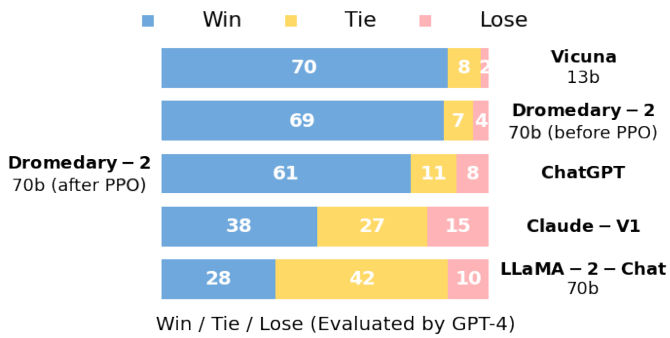
RL training The RL training uses the same collection of unlabeled prompts as the `Self-Align` SFT stage, with additional 7.5k math problem prompts from the MATH [79] to improve the mathematical solving capability of our model.

Training Details

The architecture of the reward model is the same as the base `LLaMA` model, except that the embedding output of the last token is linearly projected to a scalar value to indicate the reward of the whole response. Following Dubois et al. [57], we initialize the value model from the reward model. To fit all the models (i.e., policy, reward, value, original policy) into one GPU, we adopt QLoRA [51, 81] for all the fine-tuning processes in Self-Align and SALMON. We use Proximal Policy Optimization (PPO; Schulman et al. [169]) with a KL penalty for the RL training. Experiments with non-RL (or offline RL) alternative to PPO [72, 158, 232] are left for future work. More details can be found in Appendix.

Baseline Models

Due to the space limit, we describe the details of the baseline models in the appendix. Notably, we mainly compare with non-distilled models that are aligned from scratch. While there are potentially stronger open-source LLMs, such as `Orca` [132] and `WizardLM` [217], our primary open-source baseline for comparison is `LLaMA-2-Chat` [194], as it stands out as the best open-source LLM that has been aligned from scratch.



	MT	T-1	T-2
GPT-4	9.00	8.96	9.03
ChatGPT	7.94	8.08	7.81
Claude-V1	7.90	8.15	7.65
Dromedary-2-70b	**7.37**	**7.77**	**6.96**
Vicuna-33b (KD)	7.13	7.46	6.79
Dromedary-2-70b (before PPO)	6.91	7.48	6.34
LLaMA-2-Chat-70b	6.88	7.04	6.73
Guanaco-33b	6.53	6.88	6.18
	...		

Figure 3.4: GPT-4-based automatic evaluation on Vicuna-Bench and MT-Bench. Dromedary-2 outperforms LLaMA-2-Chat-70b and thus represents the state-of-the-art chatbot performance in non-distilled open-source models.

3.5.2 Benchmark Evaluations

Chatbot Evaluation Human evaluation is often regarded as the gold standard for judging AI chatbots, but is not always scalable and reproducible. In this work, we primarily investigate automatic evaluation leveraging GPT-4 on prevalent chatbot benchmarks, deferring human evaluation to future work. In this work, we conduct GPT-4-based automatic evaluation on Vicuna-Bench [34] and MT-Bench [235] to measure the chatbot capability of our model. The results can be found in Figure 3.4. We also evaluate our model on the AlpacaEval leaderboard [110] and report the results in Table in the appendix.

General Capability Evaluation We use Big Bench Hard (BBH; Suzgun et al. [185]) as a testbed for reasoning ability, HumanEval [31] for coding ability, and TydiQA [41] for multilingual ability. We adopt the same evaluation protocol as Wang et al. [205]. The results are reported in Table 3.2 (left), where Dromedary-2 significantly outperforms the state-of-the-art open-source model, LLaMA-2-Chat.

Truthfulness Evaluation The TruthfulQA benchmark [116] evaluates a model's ability to identify true claims, specifically in the context of literal truth about the real world. We use the same few-shot evaluation protocol and decoding strategy as in Touvron et al. [194] and report the percentage of generations that are both truthful and informative, evaluated by a fine-tuned GPT-3 model, i.e., a "GPT-judge". We present the results in Table 3.2 (right), where

Table 3.2: Evaluating the general capabilities and truthfulness of the LLM-based AI agents. Big-Bench Hard (BBH), HumanEval, and TydiQA are used to evaluate **reasoning**, **coding**, and **multilingualism**, respectively. † denotes the results are taken from Wang et al. [205], where their BBH dataset is sub-sampled so may not be directly comparable. ‡ denotes the results taken from Touvron et al. [194], where their GPT-3 judge model may not be exactly the same as ours.

	BBH Direct	BBH CoT	HumanEval P@1	TydiQA GP
GPT-4†	50.9	88.0	85.7	70.8
ChatGPT†	49.0	66.1	72.2	51.9
Dromedary-2-70b	51.4	**66.3**	**40.6**	**64.3**
LLaMA-2-Chat-70b	43.1	52.2	35.0	27.9
LLaMA-2-70b	**53.1**	57.7	31.5	63.5
Vicuna-33b (KD)	41.2	50.8	21.1	37.5

	Truthful	Tru*Inf
Dromedary-2-70b	**0.98**	**0.84**
Vicuna-13b (KD)	0.84	**0.84**
ChatGPT	0.81	0.80
Dromedary-2-70b (before PPO)	0.89	0.75
LLaMA-2-Chat-70b‡	-	0.64
LLaMA-2-70b‡	-	0.50

Dromedary-2 achieves new state-of-the-art on this benchmark.

3.5.3 Improved Controllability by Principle Intervention

As a proof of concept, we demonstrate that by leveraging different principles as preference guidelines, we can fine-tune the policy model to selectively exhibit enhanced helpfulness, honesty, or harmlessness. We also show that we can define customized principles to reduce the occurrence of false refusals seen in certain over-aligned language models such as LLaMA-2-Chat [194]. Due to the space limit, please refer to Appendix for the detailed results.

3.6 Conclusion & Discussion

In this work, we introduce SALMON, a new AI alignment paradigm where a principle-following reward model is trained to effectively and flexibly align language models with human values and intentions. During the RL training stage, by merely adjusting the principles that the reward model follows, we can gain full control over the preferences of the reward model, and subsequently influence the behavior of the RL-trained policy model. This eliminates the traditional reliance on the exhaustive collection of online human preferences. Combined with the Self-Align technique [182], we build a powerful AI-assistant agent, Dromedary-2, with only six exemplars for in-context learning and 31 human-defined principles. Our self-aligned AI agent significantly surpasses the performance of several state-of-the-art RLHF-trained AI systems in chatbot, reasoning, coding, multilingualism, and truthfulness benchmarks.

Part II

Aligning Language Models Towards Truth Seeking

Chapter 4

Improving Truthfulness of Language Models with Recitation Augmentations

The previous chapters explored how large language models (LLMs) can be aligned with human values and intentions through principle-driven prompting and reinforcement learning with AI feedback (RLAIF). These efforts aimed to shaping model behavior around human-defined norms. However, being aligned with human preferences does not necessarily guarantee factual correctness. A model might generate responses that are persuasive or well-liked, yet still factually incorrect—a problem known as hallucination.

In this chapter, we shift our focus from aligning models with what humans prefer to aligning them with what is true. We propose a new paradigm, RECITation-augmented gEneration (RECITE), to improve the factual accuracy of LLMs without relying on external retrieval mechanisms. Instead, RECITE encourages models to first internally recite relevant knowledge from their own parameters, before producing final answers. This approach helps bridge the gap between what a model "knows" and what it says—paving the way for more trustworthy and grounded language generation.

4.1 Introduction

Large language models (LLMs) have achieved impressive in-context few-shot performance on knowledge-intensive NLP tasks [23, 36, 80, 157]. For example, in open-domain question answering [29], demonstrated by only a few examples of question-answer pairs, LLMs are able to answer arbitrary factoid questions [88, 98, 221]. Recent research [73, 85, 106] shows that retrieval-augmentation can further improve LLMs' performance on knowledge-intensive tasks

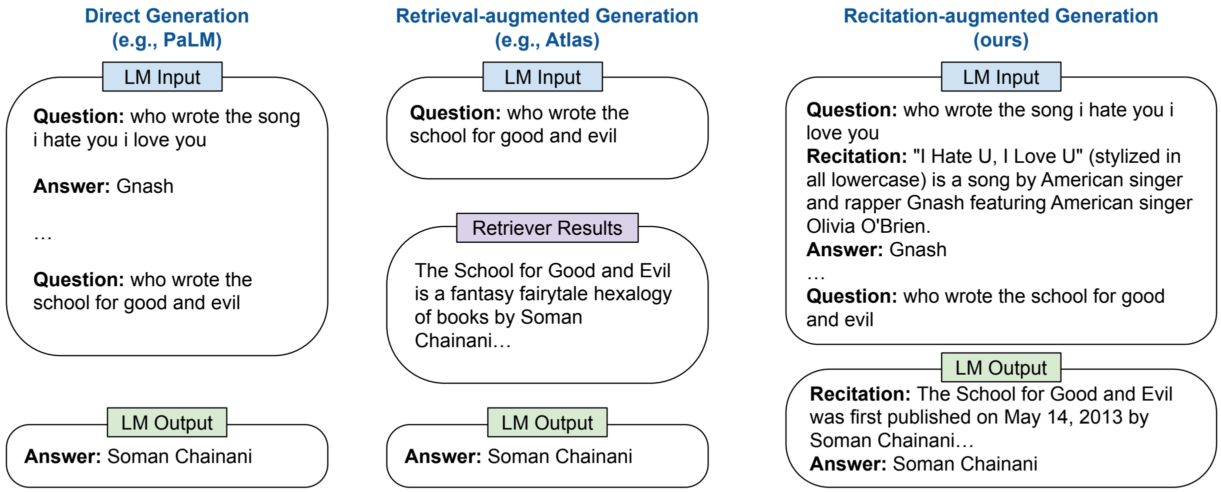


Figure 4.1: Illustration of evaluating (few-shot) open-domain question answering with (closed-book) direct generation [36], (open-book) retrieval-augmented generation [85], and (closed-book) recitation-augmented generation (ours).

by conditioning the LLMs on retrieved relevant passages from an external corpus.

This paper proposes a new paradigm to help LLMs generate more accurate factual knowledge without retrieving from an external corpus, called RECITation-augmented gEneration (RECITE), wherein we tackle knowledge-intensive NLP tasks by first reciting relevant information and then generating the outputs. Such a two-step paradigm decomposes the original knowledge-intensive task into two sub-tasks: knowledge-recitation and task-execution, where the former can be regarded as a form of intermediate knowledge retrieval step (from the model weights), while the latter is the execution step that produces the final outputs.

The motivation of introducing an additional knowledge-recitation step comes from our observation that while few-shot prompting can help LLMs execute specific NLP tasks, these tasks are usually not in a similar form as the original causal language modeling pre-training objective. This hinders LLMs from effectively reciting knowledge from their memory [26]. Consider a student taking a closed-book exam that contains knowledge-intensive questions, for example, **"what is the tenth decimal of π?"**. They typically cannot directly answer this question because in studying stage (in analogy to the language modeling pre-training stage for LLMs), it is highly unlikely that they would read "the tenth decimal of π is 5". However, there can be some sentences like "the first N digits of π are 3.14159 26535..." existing in the textbook that can be recited by the student. Therefore, a student can possibly answer this question in a recite-and-answer scheme: **"The first 10 digits of π are 3.14159 26535. So the answer is 5"**.

Here, the knowledge-recitation step can serve as an intermediate step that mimics the language modeling pre-training task, and thus better helps the LLM to generate factual knowledge.

We verify the effectiveness of our recitation-augmented generation on few-shot Closed-Book Question Answering (CBQA) tasks (referred as **recite-and-answer** in the CBQA context), as illustrated in Figure 4.1. CBQA is an attractive open-domain QA task in that a fully parameterized LM can generate answers directly without an external corpus or separate retrieval models [160]. We show that the proposed recite-and-answer scheme is an effective method for CBQA and compatible with other techniques for boosting few-shot performance of LLMs. We also show that, in addition to improving the few-shot in-context learning performance of RECITE-enhanced LLM, fine-tuning the pre-trained LLMs on synthetic generated question-passage pairs can further improve the recitation performance and lead to a better downstream QA accuracy.

Experiments on four large language models (PaLM [36], UL2 [190], OPT [229]), and Codex [31] show that a recite-and-answer scheme can improve performance on various types of CBQA tasks, including Wikipedia-based single-hop QA (Natural Questions, Kwiatkowski et al. 98), trivia questions (TriviaQA, Joshi et al. 88), and Wikipedia-based multi-hop QA (HotpotQA, Yang et al. 221).

4.2 Methodology Overview

The goal of this paper is to mimic a human's ability to recite relevant factoid knowledge [129] before answering knowledge-intensive questions, such that these questions can be answered more accurately. In the following we describe our recite-and-answer scheme for few-shot closed-book question answering (CBQA), which consists of two components: (1) a evidence-recitation module for reciting relevant passages, and (2) a question-answering module for generating answers given the recited evidence. Notice that in this paper, we focus on few-shot setting, in which we assume only a few question-answer demonstrations are provided. In Natural Questions [98] benchmark, since the questions are from queries issued to the Google search engine by multiple users, and thus can be regarded as unannotated data, we further assume that we have top-retrieved Wikipedia pages for these questions. The paragraphs in these top-retrieved Wikipedia pages will be used to generate synthetic paired question-recitation data for fine-tuning the LM (described in Section 4.4).

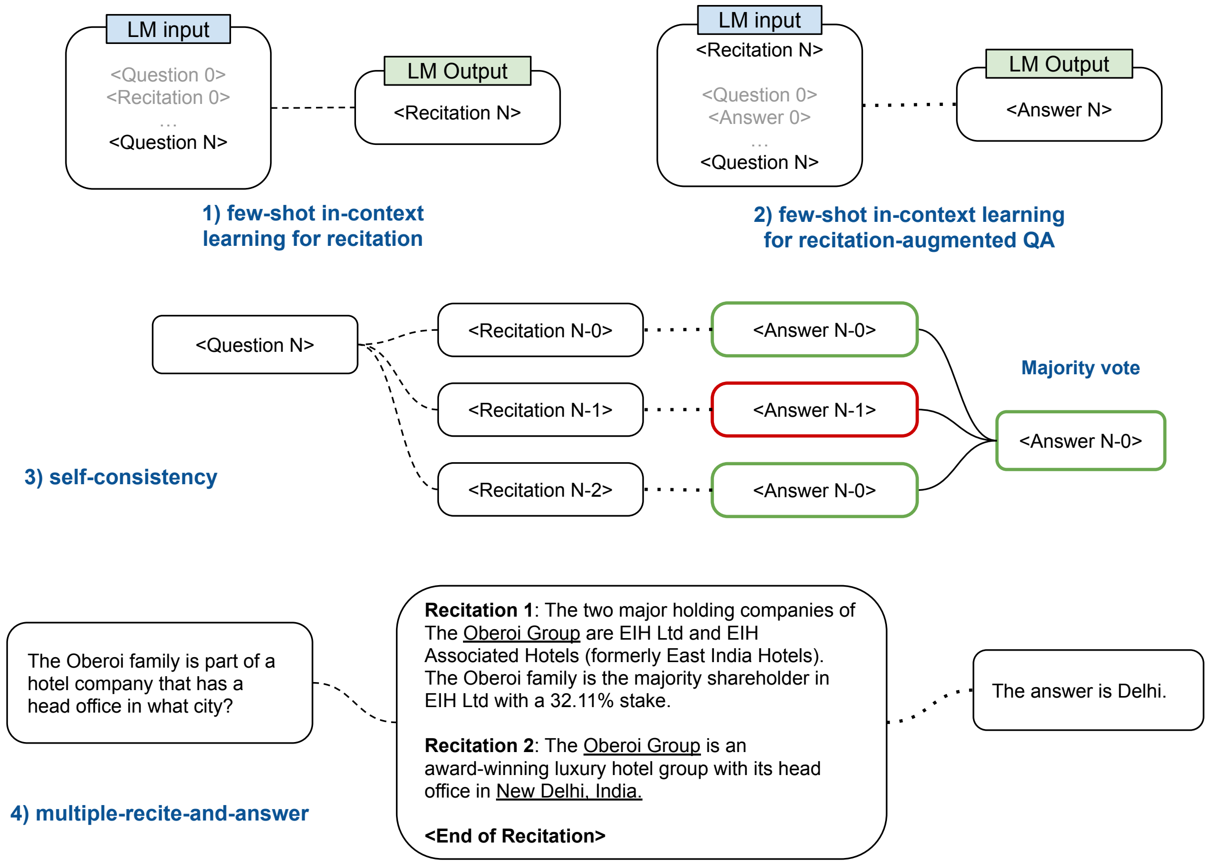


Figure 4.2: Illustration of prompt-based in-context learning for recitation generation, recitation-augmented question answering, self-consistency ensembling, and multiple-recite-and-answer for multi-hop questions (Sec. 4.3). In multiple-recite-and-answer scheme, the latter recitaiton can utilize the information from the previous ones, such as “Oberoi Group” in this case. The prompts for self-consistency and multi-hop recite-and-answer are dropped for brevity.

4.3 Prompt-based Recite-and-Answer for Question-Answering

Recitation-augmented question answering We start with single-hop question answering [88, 98], where the answers are usually supported by a specific document in the corpus, which is sometimes referred as evidence [88]. Different from chain-of-thought prompting [207] where a rationale is directly generated to explain the generated answer [88, 101, 135], we propose to first recite a passage about the question, and then answer the question based on the recitation.

We propose a prompt-based learning-to-recite scheme by leveraging the LLM’s in-context learning ability [23]. We prompt the LLM with paired exemplars of questions and recited evidences, and the LLM can learn in an in-context manner to generate a recitation for an arbitrary question. To perform recitation-conditioned few-shot question answering, we append the recited passages at the beginning of the original question-answer exemplars as a single prompt, and then generate the final answer (Step 1 & 2 in Figure 4.2).

Self-consistency ensemble The factual knowledge about a question can appear in several places in the language model's training corpora. For example, the fact of "Queen Elizabeth II opened the London Bridge on 17 March 1973" can appear in both Wikipedia page "London Bridge" and page "March 1973", so it is highly likely that there exists knowledge from different articles that could lead to the same, correct answer. With this motivation, we argue that similar to multi-step reasoning in chain-of-thought, recitation-augmented question answering can also benefit from the self-consistency technique with multiple-path decoding [203]. Specifically, given an arbitrary question, we first use top-k sampling to independently generate a few recitations, and then greedy decode the answer of the question based on the sampled recitations. Finally, we determine the optimal answer by taking a plurality/majority vote (Step 3 in Figure 4.2).

Multiple-recite-and-answer for multi-hop question-answering Multi-hop question answering requires the QA system to find and reason over multiple supporting documents. However, the nature of recitation restricts us to recite passages from one article at a time. In order to apply the recite-and-answer scheme to solve multi-hop questions, we introduce multiple-recite-and-answer scheme (Step 4 in Figure 4.2), that is, given the multiple-hop question, we use the prompt words such as "Recitation 1" and "Recitation 2" to elicit the LLM to generate recitation passages on different topics. Since the multiple recited passages are generated in one-pass from the LLM decoding sequentially, the generation of later passages can effectively utilize the information both in the original question and the previous recited ones. Our multiple-recite-and-answer scheme for multi-hop question-answering is also compatible with the self-consistency technique, by applying top-k sampling when generating multiple recitations and performing majority voting for the final answers.

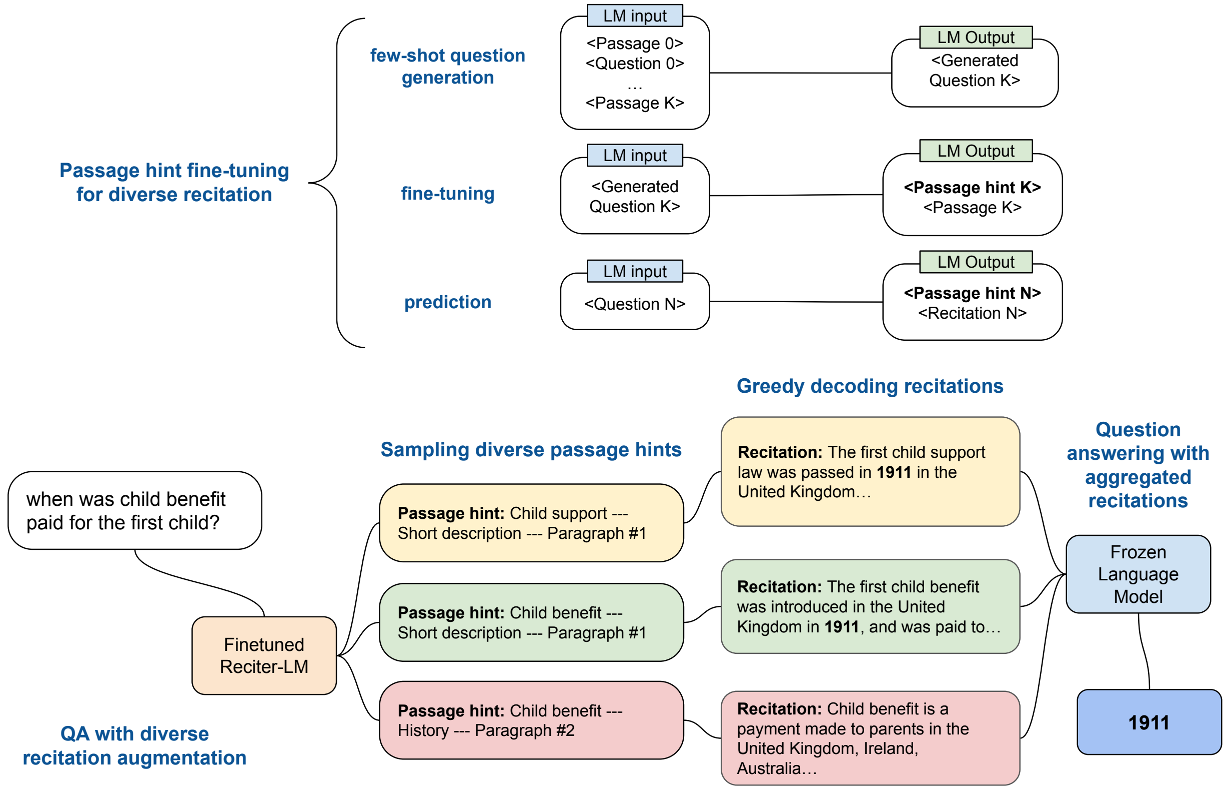


Figure 4.3: Illustration of question answering with diverse recitation and the corresponding few-shot question generation and fine-tuning processes.

4.4 Passage Hint-based Diversified Recitation with Fine-Tuning

Passage hint-based diversified recitation While the sampling-based recitation and self-consistency improves the robustness of recite-and-answer method, one argument for its inefficiency is that if the evidence-recitation module samples the wrong facts about the question, the question-answering module will not be able to figure it out and tend to generate the wrong answer. Therefore, on the one hand, we need to use a low sampling temperature to avoid generating recitations with wrong facts, on the other hand, we want to make sure the sampled recitations have enough diversity.

To tackle such a dilemma, we propose *passage hint-based diversified recitation*. We observe that in well-formed text knowledge bases, such as Wikipedia, we can usually find a unique passage hint for each passage, by concatenating the section titles and the in-section order of each passage. For example, the passage hint of the second passage in Section 5.2 "Enforcement" of Wikipedia page "Child support" would be "Child support — Compliance and enforcement issues — Enforcement — Paragraph #2". In passage hint-based diversified recitation, we first use *sampling* to generate a diverse set of passage hints, and then use *greedy decoding* to ensure the factual accuracy of the contents in each passage.

Since each passage hint corresponds to a unique passage, we can first de-duplicate the passage hints and then generate the full passages to get more diverse recitation passages. Furthermore, as the recited passages are less likely to be similar due to unique passage hints, inspired by recent progress on question-answering with multiple retrieved passages [84], we use aggregated diverse recitations as a single context, and generate the answer with a few more question-answer pair demonstrations. Figure 4.3 (lower) illustrates the recite-and-answer scheme with passage hint-based diversified recitation.

Fine-tuning on few-shot generated questions We found that although the training data of many existing LLMs [36, 53] contains the Wikipedia corpora, which are usually regarded as the factoid documents for knowledge-intensive question answering tasks [88, 98], the section titles are usually not explicitly included in training. This makes the pre-trained LLM hard to discover the mapping from the question to the passage hint, and to the full passage merely by few-shot prompting.

To address this issue, we propose an additional fine-tuning stage to adapt LLMs to learn such mappings. Assuming we have access to not only a few question-answer pairs, but also the top-retrieved Wikipedia pages for queries issued to the Google search engine by multiple users [98], we can use few-shot prompting to generated synthetic question-hint-passage pairs and then finetune the LLMs on the generated data.

Specifically, we use the ground-truth evidence and question pairs as the prompt, and generate new questions with in-context learning for randomly sampled passages from Wikipedia pages. Next, based on the few-shot generated questions, we train the LLM to predict the original passage hint, as well as the passage content. Figure 4.3 (upper) illustrates the whole process of passage hint fine-tuning.

4.5 Experiments

In this section, we report empirical evaluations of our proposed RECITE with recite-and-answer schemes on a diverse set of few-shot closed-book question answering tasks and different language models with varying scales.

4.5.1 Experimental setup

Evaluation Datasets

Natural Questions Natural Questions [98] consists of questions aggregated from the Google search engine and the answers from the Wikipedia page in the top 5 search results. We treat it as a single-hop question answering task. Since Natural Questions contains the so-called "long answer" annotations, which is a whole HTML bounding box containing enough information to infer the answer, we directly utilize the "long answer" as the ground-truth recitation exemplars in our prompt (Sec. 4.3). In order to make a direct comparison with recent LLMs [36, 85], we evaluate our methods in 5-shot and 64-shot settings.

TriviaQA TriviaQA dataset [88] is constructed by collecting Trivia enthusiast authored question-answer pairs and their retrospectively collected evidence. Since there is no obvious way to collect a "long answer" in the retrospective evidence documents (the exact appearances of the answer may contain enough information to infer the answer), we evaluate TriviaQA in the single-hop 5-shot setting, and manually compose the recitation passage from Wikipedia for 5 randomly sampled training questions. The concrete prompt can be found in the appendix.

HotpotQA HotpotQA [221] is designed to explicitly test QA systems' ability to perform multi-hop reasoning. It is collected by explicitly composing questions requiring reasoning about multiple supporting context documents. Following Wang et al. [202], we evaluate HotpotQA as a multi-hop question answering task in the 4-shot setting. But instead of chain-of-thought prompting as in [202], we use multiple-recite-and-answer (Sec. 4.3) to achieve multi-step reasoning. We also provide the concrete prompt in the appendix.

Metrics We calculate the Exact Matching (EM) and F1 scores for the normalized answers, while the specific text normalization applied on each dataset can be slightly different.

Pre-trained language models

We evaluate the effectiveness of RECITE on four langauge models: PaLM, UL2 [190], OPT [229], and Codex [23, 31, 144]. Due to the space limit, the detailed descriptions of them are provided in Appendix.

Table 4.1: Performance comparison on Natural Questions (NQ), TriviaQA, and HotpotQA. The number of shots for each task are mentioned in parenthesis.

		PaLM-62B EM / F1	UL2-20B EM / F1	OPT-30B EM / F1	Codex-002 EM / F1
NQ	Standard-prompting (direct)	25.76 / 36.47$_{(5)}$ 28.98 / 40.13$_{(64)}$	10.16 / 20.17$_{(5)}$ 12.70 / 21.97$_{(16)}$	14.97 / 22.93$_{(5)}$	31.45 / 44.75$_{(5)}$
	Recite-and-answer (20-path)	**28.70** / **39.76**$_{(5)}$ **31.34** / **42.48**$_{(64)}$	**14.16** / **23.13**$_{(5)}$ **14.94** / **24.29**$_{(16)}$	**17.84** / **26.74** $_{(5)}$	**35.84** / **49.12**$_{(5)}$
TriviaQA	Standard-prompting (direct)	65.38 / 71.85$_{(5)}$	48.73 / 54.32$_{(5)}$	45.90 / 50.68$_{(5)}$	81.84 / 86.09$_{(5)}$
	Recite-and-answer (20-path)	**65.84** / **72.10**$_{(5)}$	**53.42** / **58.69**$_{(5)}$	**49.02** / **54.22**$_{(5)}$	**83.50** / **88.03**$_{(5)}$
HotpotQA	Standard-prompting (direct)	20.51 / 28.90$_{(4)}$	16.99 / 24.99$_{(4)}$	16.70 / 25.21$_{(4)}$	28.32 / 39.03$_{(4)}$
	Chain-of-thought (20-path)	23.73 / 32.80$_{(4)}$	17.68 / 24.87$_{(4)}$	16.89 / 24.03$_{(4)}$	34.38 / 45.50$_{(4)}$
	Recite-and-answer (20-path)	**26.46** / **35.67**$_{(4)}$	**19.04** / **27.32**$_{(4)}$	**17.77** / **26.58**$_{(4)}$	**37.11** / **48.37**$_{(4)}$

4.5.2 Experiments

We use the test split for all tasks if the test split is available and has labels for evaluation, otherwise we use the dev split. In addition, TriviaQA and HotpotQA are too large to run large language models on, so we used the first 1,024 data points for evaluation.

Prompt-based results

We report the single-hop closed-book question answering (CBQA) evaluation results on Natural Questions (NQ) and TriviaQA and the multi-hop CBQA evaluation results on HotpotQA. In Tab. 4.1, we report the results with prompt-based in-context learning and self-consistency.

From the tables, we can see that the proposed recite-and-answer scheme can significantly improve the CBQA performance on both datasets with various pre-trained language models. While the performance improvements on NQ is more consistent across different language models, we find that the improvements from recite-and-answer is more significant on smaller language models on TriviaQA. Our hypothesis is that the Trivia-style question usually contains more contextual information in the question, thus weakened the effectiveness of recitation for strong LLMs like PaLM.

Besides, we can see that the recite-and-answer scheme can outperform the chain-of-thought prompting performance on the multi-hop reasoning task. Interestingly, we also find that for LLMs that have large gains from chain-of-thought (i.e., PaLM), they also have large improvements from recite-and-answer.

Table 4.2: Performance comparison of PaLM-62B on Natural Questions (NQ) dataset with standard-prompting, recite-and-answer with self-consistency sampling, and recite-and-answer with diversified recitation. The number of shots for each task are mentioned in parenthesis.

	EM / $F1_{(5)}$	EM / $F1_{(64)}$
Standard-prompting (direct)	25.76 / 36.47	28.98 / 40.13
Recite-and-answer (20-path)	28.70 / 39.76	31.34 / 42.48
Recite-and-answer w/ diversified recitation (20-path)	**32.20** / **44.02**	**33.23** / **45.29**

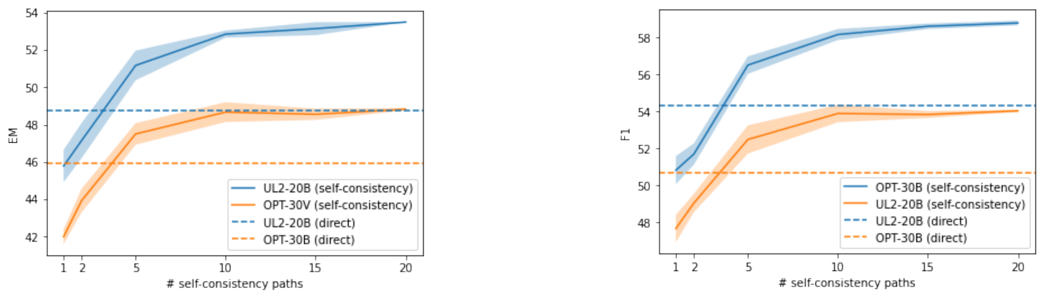


Figure 4.4: TriviaQA EM/F1 on OPT-30B and UL2-20B with different # of self-consistency paths.

Results of passage hint-based diversified recitation

For Natural Questions dataset, since it has the collection of top-retrieved Wikipeida pages corresponding to the unannotated queries issued to the Google search engine, we additionally report the diversified recitation results of fine-tuned PaLM model in Tab. 4.2. From the table, we find that diversified recitation can further improve the performance of PaLM on the NQ dataset.

4.5.3 Analysis

On the number of self-consistency paths

We analyze how the number of passages recited would affect the performance of recite-and-answer under the self-consistency setting. Due to the costly inference of LLMs, we first sample up to $k = 20$ recitation passages, and then apply self-consistency to a randomly selected subset of recitations to simulate less paths. For each number of self-consistency paths, we evaluate the randomly selected subsets five times and report the mean and standard deviation. We conduct an analysis on OPT-30B and UL2-20B on the TriviaQA dataset and report the results in Fig. 4.4. We can see that sampling more recitation passages tends to improve the recite-and-answer

Table 4.3: Natural Questions (NQ) results with different context passages.

	$\text{UL2-20B}_{(5)}$ EM / F1	$\text{Codex-002}_{(5)}$ EM / F1
No passage	10.16 / 20.17	31.45 / 44.75
Ground-truth passage	41.02 / 55.73	49.32 / 64.32
BM25-Retrieval (Top-1)	16.31 / 27.66	33.20 / 47.45
$\text{LM-Recitation}_{(5)}$ (20-path)	14.16 / 23.13	35.84 / 49.12

Table 4.4: Per-question error analysis on TriviaQA.

	$\text{UL2-20B}_{(5)}$	$\text{OPT-30B}_{(5)}$
Hits@Majority	53.42%	49.02%
Not Recit.	21.09%	22.27%
Hits@20-Recit.	5.66%	8.01%
Hits@20-Path	19.82%	20.07%

Table 4.5: Per-path error analysis on TriviaQA.

Recit.	Ans.	$\text{UL2-20B}_{(5)}$	$\text{OPT-30B}_{(5)}$
✓	✓	33.60%	30.06%
✓	✗	7.87%	9.79%
✗	✓	12.10%	12.57%
✗	✗	46.44%	47.58%

performance, while less randomness is observed with more self-consistency paths.

On the robustness of few-shot exemplars

A well-known problem of in-context few-shot learning is its instability to the choices of exemplars and their orders [233]. We evaluate the robustness of standard prompting and our recite-and-answer method with 5 random seeds and report the mean and standard deviation of UL2 model running on the TriviaQA dataset in Tab. The 5-shot exemplars are randomly sampled and shuffled for each seed. From the table, we can see that with recitation sampling, recite-and-answer exhibits similar robustness (in terms of small performance deviation) as standard prompting under different random seeds and numbers of self-consistency paths. The overall gains by recite-and-answer are significant compared to standard prompting regardless of the choice of few-shot exemplars.

Recitation v.s. Retrieval v.s. Ground-truth

One may ask without the external corpus, whether the quality of recited passages with LLMs is better than simple retrieval models, e.g., BM25 [161][1]. To answer this question, we evalu-

[1]We use the pre-indexed "enwiki-paragraphs" corpus in the pyserini package (https://github.com/castorini/pyserini), which is originally designed for BERTserini [220].

ate the few-shot question-answering performance of UL2 and Codex on three kinds of context passages: retrieval, recitation, and ground-truth. We report the results on first 1024 validation examples in Natural Questions (NQ) dataset, since it is the only dataset that contains the “long answer” annotation that can be regarded as ground-truth context passage. From Tab. 4.3, we can see that the classic retrieval model, i.e., BM25, is still a very strong baseline for collecting information from the corpus. Nonetheless, compared to BM25, our recite-and-answer still achieves a quite competitive performance via generation only and without using any external corpus. Besides, we find that stronger models (i.e., Codex) tend to benefit more from the the model’s own recitation than BM25 retrieved context.

Error analysis

We perform an error analysis on the 1024 evaluation examples in the TriviaQA dataset. We classify the errors into three categories: 1) Not Recit., i.e., the correct answer is not recited in any of the 20 recited passages in self-consistency. 2) Hits@20-Recit., i.e., the correct answer can be found in one of the recited passage, but does not appear in the QA module’s outputs. 3) Hits@20-Path, i.e., the correct answer is one of the final outputs of the 20 self-consistency paths, but it does not have the majority votes. The correct final answer is marked as Hits@Majority (i.e., Exact Matching). An algorithmic description is given in Algo in appendix. We report the results of UL2-20B and OPT-30B in Tab. 4.4. We can see that “No Recit” and “Hits@20-Path” account for the majority of the errors, meaning that the QA module performs quite well (if the correct answer appears in one of the recitation passages, it will be extracted by the QA module in most of the cases), and the main bottleneck still lies in the recitation quality and answer aggregation strategies.

We also perform a per-path error analysis, i.e., how many questions can be answered correctly (or not) when the recitation exactly contains (or not) the answer tokens. The results are shown in Tab. 4.5. We can see that around $7\% \sim 10\%$ questions have the correct recitation but cannot produce the correct answer, while around 12% questions do not have the correction recitation but can be answered correctly anyway.

4.6 Conclusion & Discussion

In this paper, we propose a novel recitation-augmented generation framework to improve language models’ performance in the closed-book question-answering setting. We hypothesize that for knowledge-intensive NLP tasks, encouraging the model to explicitly recite a specific knowledge source would be helpful in augmenting its memory. In addition, we found that diversifying the recitation process can be beneficial as well since usually there exists multiple knowledge sources that could be used to answer the same question. We show promising results over three large language models and across three different closed-book QA datasets, demonstrating the effectiveness of our proposed recite-and-answer approach.

One limitation of our method is that updating time-sensitive knowledge for a pure LLM-based method requires training or fine-tuning the LLMs on the new corpus, which can be costly. For future work, we plan to further validate the effectiveness of recitation-augmented generation for other knowledge-intensive NLP tasks in the closed-book setting, such as fact checking.

Chapter 5

Aligning Multimodal Models with Factually Augmented RLHF

In the previous chapter, we focused on improving the factual accuracy of large language models (LLMs) by encouraging them to recite internal knowledge before answering—a process that enhanced truthfulness in purely textual settings. However, as models expand beyond text and become capable of interpreting images and other modalities, new challenges arise. The problem of hallucination persists, but now it manifests in more complex ways: a model might describe objects that aren't in an image, infer context incorrectly, or generate confident but false visual claims. In this chapter, we extend the pursuit of truthfulness to Large Multimodal Models (LMMs) by aligning them not just with human preferences, but also with factual visual grounding. We introduce a novel approach called Factually Augmented RLHF, which adapts Reinforcement Learning from Human Feedback to the multimodal setting and incorporates ground-truth signals (such as captions and structured answers) to improve the model's fidelity and reduce hallucinations.

5.1 Introduction

Large Language Models (LLMs; Brown et al. [23], Chowdhery et al. [36], OpenAI [141]) can delve into the multimodal realm either by further pre-training with image-text pairs [3, 13] or by fine-tuning them with specialized vision instruction tuning datasets [121, 243], leading to the emergence of powerful Large Multimodal Models (LMMs). Yet, developing LMMs faces challenges, notably the gap between the volume and quality of multimodal data versus text-only datasets. Consider the LLaVA model [121], which is initialized from a pre-trained vision encoder

Question:
Where is this photo taken?

Ground Truth:
From the slogan "Welcome to Houston, Texas", we can infer that this photo is taken at an airport in Houston, Texas.

LLaVA:
This photo is taken at an airport.

LLaVA-SFT$^+$:
This photo is taken at the Houston airport.

LLaVA-RLHF :
This photo is taken in the baggage claim area of an airport, specifically in the lobby of the George Bush Intercontinental Airport in Houston, Texas. The large sign in the background indicates the airport's name and location. The presence of multiple people, handbags, and suitcases in the scene further supports the fact that this is indeed an airport setting.

Question:
What is the gender of the person riding the motorcycle in the image?

Ground Truth:
No one is riding the motorcycle in the image.

LLaVA:
The person riding the motorcycle in the image is a woman.

LLaVA-SFT$^+$:
The person riding the motorcycle in the image is a man.

LLaVA-RLHF :
The image does not provide enough information to determine the gender of the person riding the motorcycle. It only shows a motorcycle parked on the side of the road, and there is no indication of the rider's gender. The rider could be a man or a woman, or even someone who identifies as non-binary or gender non-conforming. It is important to avoid making assumptions about a person's gender based on visual cues alone, as these assumptions can be inaccurate and potentially harmful.

Table 5.1: Qualitative examples to illustrate the effect of RLHF on Large Multimodal Model. LLaVA-RLHF is presented to be more helpful (above) and less hallucinated (bottom).

[156] and an instruction-tuned language model [34]. It is trained on just 150K synthetic image-based dialogues, which is much less in comparison to the text-only models (Flan [124] utilizing over 100M examples spanning 1800 tasks. Such limitations in data can lead to misalignment between the vision and language modalities. Consequently, LMMs may produce hallucinated outputs, which are not accurately anchored to the context provided by images.

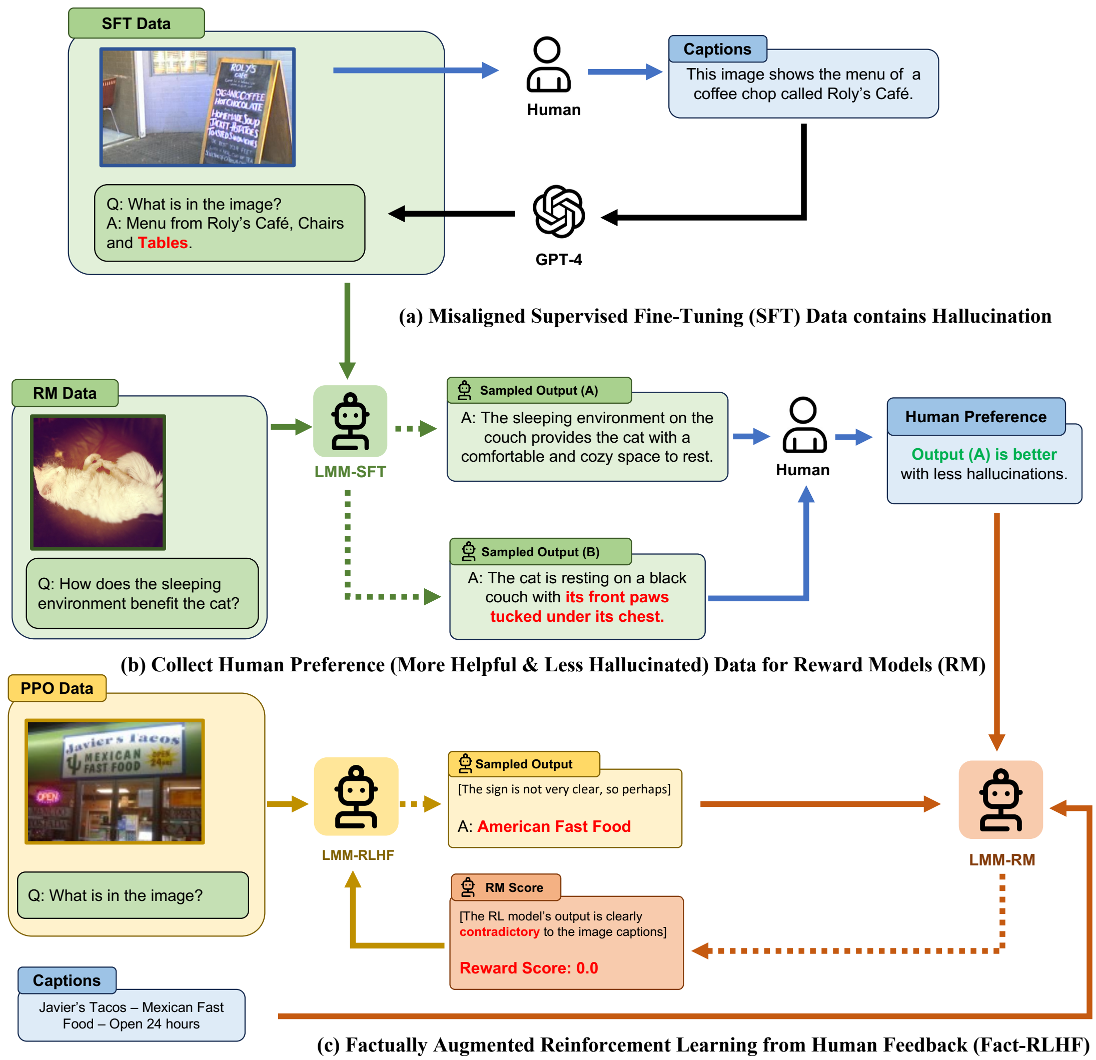


Figure 5.1: Illustration of how hallucination may occur during the Supervised Fine-Tuning (SFT) phase of LMM training and how Factually Augmented RLHF alleviates the issue of limited capacity in the reward model which is initialized from the SFT model.

To mitigate the challenges posed by the scarcity of high-quality visual instruction tuning data for LMM training, we introduce **LLaVA-RLHF**, a vision-language model trained for improved multimodal alignment. One of our key contributions is the adaptation of the Reinforcement Learning from Human Feedback (RLHF) [16, 144, 178], a general and scalable alignment paradigm that shows great success for text-based AI agents, to the multimodal alignment for LMMs. By collecting human preferences with an emphasis on detecting hallucinations[1], we uti-

[1]We instructed crowdworkers to prioritize the responses that exhibit better multimodal alignment and minimize hallucinations. That is, if two responses are free of hallucinations, the crowdworkers were asked to choose a

lize those preferences in reinforcement learning for LMM fine-tuning [178, 244]. Our approach can improve the multimodal alignment with a relatively low annotation cost, e.g., collecting 10K human preferences for image-based conversations with $3000. To the best of our knowledge, this approach is the first successful adaptation of RLHF to multimodal alignment.

A potential issue with the current RLHF paradigm is called *reward hacking*, which means achieving high scores from the reward model does not necessarily lead to improvement in human judgments. To prevent reward hacking, previous work [16, 194] proposed to iteratively collect "fresh" human feedback, which tends to be costly and cannot effectively utilize existing human preference data. In this work, we propose a more data-efficient alternative, i.e., we try to make the reward model capable of leveraging existing human-annotated data and knowledge in larger language models. Firstly, we improve the general capabilities of the reward model by using a better vision encoder with higher resolutions and a larger language model. Secondly, we introduce a novel algorithm named **Factually Augmented RLHF (Fact-RLHF)**, which calibrates the reward signals by augmenting them with additional information such as image captions or ground-truth multi-choice option, as illustrated in Fig. 5.1.

To improve the general capabilities of LMMs during the Supervised Fine-Tuning (SFT) stage, we further augment the synthetic vision instruction tuning data [121] with existing high-quality human-annotated multi-modal data in the conversation format. Specifically, we convert VQA-v2 [69] and A-OKVQA [171] into a multi-round QA task, and Flickr30k [225] into a Spotting Captioning task [30], and train the **LLaVA-SFT$^+$** models based on the new mixture of data.

Lastly, we look into assessing the multimodal alignment of LMMs in real-world generation scenarios, placing particular emphasis on penalizing any hallucinations. We create a set of varied benchmark questions that cover the 12 main object categories in COCO [118] and include 8 different task types, leading to MMHal-Bench. Our evaluation indicates that this benchmark dataset aligns well with human evaluations, especially when scores are adjusted for anti-hallucinations. In our experimental evaluation, as the first LMM trained with RLHF, LLaVA-RLHF delivers impressive outcomes. We observed a notable enhancement on LLaVA-Bench, achieving 94%, an improvement by 60% in MMHal-Bench, and established new performance benchmarks for LLaVA with a 52.4% score on MMBench [123] and an 82.7% F1 on POPE [111].

more helpful one.

5.2 Methodology Overview

In this study, we employ a multimodal Reinforcement Learning from Human Feedback (RLHF) approach to align Large Multimodal Models (LMMs) with human values (Sec. 5.3). The process begins with Multimodal Supervised Fine-Tuning to establish a foundational understanding of multimodal inputs (Sec. 5.3.1). This is enhanced by Multimodal Preference Modeling, where a reward model is trained with human-annotated comparisons to discern better responses (Sec. 5.3.2). The approach culminates with Reinforcement Learning and Factually Augmented RLHF, which refine the model's responses for accuracy and factual alignment, leveraging high-quality instruction-tuning data and additional ground-truth information to combat reward hacking and hallucinations (Sec. 5.4).

5.3 Multimodal RLHF

Reinforcement Learning from Human Feedback (RLHF) [16, 144, 178, 244] has emerged as a powerful and scalable strategy for aligning Large Language Models (LLMs) with human values. In this work, we use RLHF to align LMMs. The basic pipeline of our multimodal RLHF can be summarized into three stages:

Multimodal Supervised Fine-Tuning A vision encoder and a pre-trained LLM are jointly fine-tuned on an instruction-following demonstration dataset using token-level supervision to produce a supervised fine-tuned (SFT) model π^{SFT}.

Multimodal Preference Modeling In this stage, a reward model, alternatively referred to as a preference model, is trained to give a higher score to the "better" response. The pairwise comparison training data are typically annotated by human annotators. Formally, let the aggregated preference data be represented as $\mathcal{D}_{\mathrm{RM}} = \{(\mathcal{I}, x, y_0, y_1, i)\}$, where $\mathcal{I}$ denotes the image, x denotes the prompt, y_0 and y_1 are two associated responses, and i indicates the index of the preferred response. The reward model employs a cross-entropy loss function:

$$\mathcal{L}(r_{\boldsymbol{\theta}}) = -\mathbf{E}_{(\mathcal{I},x,y_0,y_1,i)\sim\mathcal{D}_{\mathrm{RM}}}\left[\log \sigma(r_{\boldsymbol{\theta}}(\mathcal{I}, x, y_i) - r_{\boldsymbol{\theta}}(\mathcal{I}, x, y_{1-i}))\right]. \tag{5.1}$$

Reinforcement Learning Here, a policy model, initialized through multimodal supervised fine-tuning (SFT) [144, 194], is trained to generate an appropriate response for each user query by maximizing the reward signal as provided by the reward model. To address potential over-optimization challenges, notably reward hacking, a per-token KL penalty derived from the initial policy model [144] is sometimes applied. Formally, given the set of collected images and user prompts, $\mathcal{D}_{\mathrm{RL}} = \{(\mathcal{I}, x)\}$, along with the fixed initial policy model π^{INIT} and the RL-optimized model $\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}$, the full optimization loss is articulated as:

$$\mathcal{L}(\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}) = -\mathbf{E}_{(\mathcal{I},x)\in\mathcal{D}_{\mathrm{RL}}, y\sim\pi^{RL}(y|\mathcal{I},x)} \left[r_{\boldsymbol{\theta}}(\mathcal{I}, x, y) - \beta \cdot \mathbb{D}_{KL}\left(\pi_{\boldsymbol{\phi}}^{\mathrm{RL}}(y|\mathcal{I}, x) \| \pi^{\mathrm{INIT}}(y|\mathcal{I}, x)\right)\right], \tag{5.2}$$

where β is the hyper-parameter to control the scale of the KL penalty.

5.3.1 Augmenting LLaVA with High-Quality Instruction-Tuning

Recent studies [194, 238] show that high-quality instruction tuning data is essential for aligning Large Language Models (LLMs). We find this becomes even more salient for LMMs. As these models traverse vast textual and visual domains, clear tuning instructions are crucial. Correctly aligned data ensures models produce contextually relevant outputs, effectively bridging language and visual gaps. For example, LLaVA synthesized 150k visual instruction data using the text-only GPT-4, where an image is represented as the associated captions on bounding boxes to prompt GPT-4. Though careful filtering has been applied to improve the quality, the pipeline can occasionally generate visually misaligned instruction data that can not be easily removed with an automatic filtering script, as highlighted in Table 5.1.

In this work, we consider enhancing LLaVA (98k conversations, after holding out 60k conversations for preference modeling and RL training) with high-quality instruction-tuning data derived from existing human annotations. Specifically, we curated three categories of visual instruction data: "Yes" or "No" queries from VQA-v2 (83k) [70], multiple-choice questions from A-OKVQA (16k) [127], and grounded captions from Flickr30k (23k) [224]. Our analysis revealed that this amalgamation of datasets significantly improved LMM capabilities on benchmark tests. Impressively, these results surpassed models [48, 103, 108] trained on datasets an order of magnitude larger than ours, as evidenced by Table in appendix and 5.4. [2]

[2]For a comprehensive breakdown of each dataset's influence, refer to Appendix.

Instruction
We have developed an AI assistant adept at facilitating image-based conversations. However, it occasionally generates what we call hallucinations, which are inaccuracies unsupported by the image content or real-world knowledge.
In this task, we request that you select the most appropriate response from the AI model based on the conversation context. When making this selection, primarily consider these two factors:

- **Honesty**: Fundamentally, the AI should provide accurate information and articulate its uncertainty without misleading the user. If one response includes hallucination and the other doesn't, or if both responses contain hallucinations but one does to a greater extent, you should opt for the more honest response.
- **Helpfulness**: In scenarios where both responses are free from hallucinations, you should opt for the more helpful one. The AI should attempt to accomplish the task or answer the question posed, provided it's not harmful, in the most helpful and engaging manner possible.

Annotation Task
Please select the better response from A and B
[IMAGE]
[CONVERSATION CONTEXT]
[RESPONSE A]
[RESPONSE B]
Question 1: Which response has fewer hallucinations in terms of the given image?
Question 2: If you have selected a tie between Response 1 and Response 2 from the previous question, which response would be more helpful or less incorrect?

Table 5.2: The instruction to the crowdworkers for human preference collection.

5.3.2 Hallucination-Aware Preference Model

Our preference model training process integrates a single reward model that emphasizes both multimodal alignment and overall helpfulness[3]. We collect human preferences on 10k hold-out LLaVA data by re-sampling the last response with our SFT model and a temperature of 0.7. The reward model is initialized from the SFT model to obtain the basic multimodal capabilities.

[3]We are considering the development of a distinct Honest reward model, inspired by the approach in Touvron et al. [194]. This introduces the possibility of constructing a piecewise Honesty-prioritized reward model. We earmark this direction for future exploration.

5.4 Factually Augmented RLHF (Fact-RLHF)

We conduct multimodal RLHF on 50k hold-out LLaVA conversations, with additional 12k multi-choice questions from A-OKVQA and 10k yes/no questions subsampled from VQA-v2. Due to the concerns of existing hallucinations in the synthetic multi-round conversation data of LLaVA, we only use the first question in each conversation for RL training, which avoids the pre-existing hallucinations in the conversational context.

Reward Hacking in RLHF In preliminary multimodal RLHF experiments, we observe that due to the intrinsic multimodal misalignment in the SFT model, the reward model is weak and sometimes cannot effectively detect hallucinations in the RL model's responses. In the text domain, previous work [16, 194] proposed to iteratively collect "fresh" human feedback. However, this can be quite costly and cannot effectively utilize existing human-annotated data and there is no guarantee that more preference data can significantly improve the discriminative capabilities of the reward model for multimodal problems.

Facutual Augmentation To augment the capability of the reward model, we propose Factually Augmented RLHF (Fact-RLHF), where the reward model has access to additional ground-truth information such as image captions to calibrate its judgment. In original RLHF [140, 178], the reward model needs to judge the quality of the response only based on the user query (i.e., the input image and prompt):

```
Image: [IMAGE]
User: [USER PROMPT]
Assistant: [RESPONSE]
Reward Model: [SCORE]
```

In Factually Augmented RLHF (Fact-RLHF), the reward model has additional information about the textual descriptions of the image:

```
Image: [IMAGE]
Factual Information: [5 COCO IMAGE CAPTIONS / 3 A-OKVQA RATIONALS]
User: [USER PROMPT]
Assistant: [RESPONSE]
Augmented Reward Model: [SCORE]
```

This prevents the reward model hacked by the policy model when the policy model generates some hallucinations that are clearly not grounded by the image captions. For general questions with COCO images, we concatenate the five COCO captions as the additional factual information, while for A-OKVQA questions, we use the annotated rationals as the factual information. The factually augmented reward model is trained on the same binary preference data as the vanilla reward model, except that the factual information is provided both during the model fine-tuning and inference.

Symbolic Rewards: Correctness Penalty & Length Penalty Certain questions come with a predetermined ground-truth answer in our RL data, including binary choices (e.g., "Yes/No") in VQA-v2 and multiple-choice options (e.g., "ABCD") in A-OKVQA. These annotations can also be regarded as additional factual information. Therefore, in the Fact-RLHF algorithm, we introduce a symbolic reward mechanism that penalizes selections that diverge from these ground-truth options. Furthermore, we observed that RLHF-trained models often produce more verbose outputs, a phenomenon also noted by Dubois et al. [57]. While these verbose outputs might be favored by users or by automated LLM-based evaluation systems [182, 235], they tend to introduce more hallucinations for LMMs. In this work, we incorporate the response length, measured in the number of tokens, as an auxiliary penalizing factor.

5.5 Experiments

5.5.1 Neural Architectures

Base Model We adopt the same network architecture as LLaVA [121]. Our LLM is based on Vicuna [34, 193], and we utilize the pre-trained CLIP visual encoder, ViT-L/14 [156]. We use grid features both before and after the final Transformer layer. To project image features to the word embedding space, we employ a linear layer. It's important to note that we use the pre-trained linear projection layer checkpoints from LLaVA, concentrating on the end-to-end fine-tuning phase for multi-modal alignment in our study. For LLaVA-SFT$^{+}{}_{7\text{B}}$, we use a Vicuna-V1.5$_{7\text{B}}$ LLM and ViT-L/14 with image resolution 256×256. For LLaVA-SFT$^{+}{}_{13\text{B}}$, we use a Vicuna-V1.5$_{13\text{B}}$ LLM and ViT-L/14 with image resolution 336×336.

Reward Model The architecture of the reward model is the same as the base LLaVA model, except that the embedding output of the last token is linearly projected to a scalar value to indicate the reward of the whole response. We use our own collected 10k human preference data to train the reward model with the cross-entropy loss (Eq. 5.1). Following Ouyang et al. [144], we train the reward model for only one epoch to avoid over-fitting (mis-calibration). A size of 500 validation data is also held out for early stopping. The final reward model's accuracy on the validation data is 65%, which is near our observed human labeler consistency of 69%

Model	Subsets			Full-Set
	Conv	Detail	Complex	
LLaVA$_{7B}$	75.1	75.4	92.3	81.0
VIGC$_{7B}$	83.3	**80.6**	93.1	85.8
LLaVA-SFT^{+}$_{7B}$	88.8	74.6	95.0	86.3
LLaVA-RLHF$_{7B}$	**93.0**	79.0	**109.5**	**94.1**
LLaVA$_{13B\times336}$	87.2	74.3	92.9	84.9
VIGC$_{13B\times336}$	88.9	77.4	93.5	86.8
LLaVA-SFT^{+}$_{13B\times336}$	85.8	75.5	93.9	85.2
LLaVA-RLHF$_{13B\times336}$	**93.9**	**82.5**	**110.1**	**95.6**

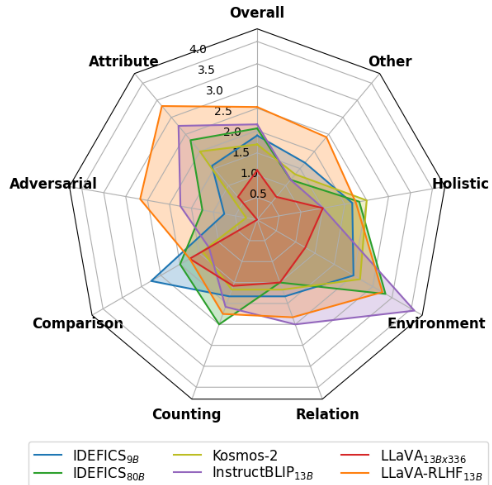


Table 5.3: (left) Automatic evaluation of LLaVA-RLHF on the LLaVA-Bench Evaluation. GPT-4 compares the answers from the VLM model outputs with the answers by GPT-4 (text-only) and gives a rating. We report the relative scores [121] of VLM models compared to GPT-4 (text-only). (right) Detailed performance of different models on the eight categories in MMHal-Bench, where "Overall" indicates the averaged performance across all categories. The questions are collected by adversarially filtering on the original LLaVA$_{13Bx336}$ model.

(Appendix).

RL Models: Policy and Value Following Dubois et al. [57], we initialize the value model from the reward model. Therefore, when training an LLaVA$_{7B}$ policy model with an LLavA$_{13B}$ reward model, the value model is also 13B. To fit all the models (i.e., police, reward, value, original policy) into one GPU, we adopt LoRA [81] for all the fine-tuning processes in RLHF. We use Proximal Policy Optimization (PPO; Schulman et al. [169]) with a KL penalty for the RL training. Without further notice, both LLaVA-RLHF$_{7B}$ and LLaVA-RLHF$_{13B}$ are trained with a LLaVA-SFT^{+}$_{13B}$ initialized reward model. More details can be found in Appendix.

5.5.2 Results

We use LLaVA-Bench [121] and our MMHal-Bench[4] as our main evaluation metrics for their high alignment with human preferences. In addition, we conducted tests on widely-recognized Large Multimodal Model benchmarks. We employed MMBench [123], a multi-modal bench-

[4]See detailed data collection for MMHal-Bench in Appendix and hallucination-aware human preference data in Appendix.

mark offering an objective evaluation framework comprising 2,974 multiple-choice questions spanning 20 ability dimensions. This benchmark utilizes ChatGPT to juxtapose model predictions against desired choices, ensuring an equitable assessment of VLMs across varying instruction-following proficiencies. Furthermore, we incorporated POPE [111], a polling-based query technique, to offer an evaluation of VLM object perception tendencies.

High-quality SFT data is crucial for capability benchmarks. By delving into the specific performances for the capability benchmarks (i.e., MMBench and POPE), we observe a notable improvement in capabilities brought by high-quality instruction-tuning data (LLaVA-SFT^{+}) in Tables 5.4. LLaVA-SFT$^{+}{}_{7B}$ model exemplifies this with an impressive performance of 52.1% on MMBench and an 82.7% F1 score on POPE, marking an improvement over LLaVA by margins of 13.4% and 6.7% respectively. However, it's worth noting that LLaVA-SFT^{+} does trail behind models like Kosmos and Shikra. Despite this, LLaVA-SFT^{+} stands out in terms of sample efficiency, utilizing only 220k fine-tuning data—a 5% fraction of what's employed by the aforementioned models. Furthermore, this enhancement isn't confined to just one model size. When scaled up, LLaVA-SFT$^{+}{}_{13Bx336}$ achieves commendable results, attaining 57.5% on MMBench and 82.9% on POPE. Comparatively, the effect of RLHF on the capability benchmarks is more mixed. LLaVA-RLHF shows subtle degradations at the 7b scale, but the LLaVA-RLHF$_{13B}$ improves over LLaVA-SFT$^{+}{}_{13B}$ by 3% on MMBench. This phenomenon is similar to the **Alignment Tax** observed in previous work [16]. Nonetheless, with our current empirical scaling law of LLaVA-RLHF [11, 91], we believe RLHF alignment would not damage the in-general capabilities of LMMs for models of larger scales.

RLHF improves human alignment benchmarks further. From another angle, even though high-quality instruction data demonstrates large gains in capability assessment, it does not improve much on human-alignment benchmarks including LLaVA-Bench and MMHal-Bench, which is also evident in recent LLM studies [205]. LLaVA-RLHF show a significant improvement in aligning with human values. It attains scores of 2.05 (7b) and 2.53 (13b) on MMHal-Bench and improves LLaVA-SFT^{+} by over 10% on LLaVA-Bench. We also presented qualitative examples in Table 5.1, which shows LLaVA-RLHF produces more reliable and helpful outputs.

Table 5.4: CircularEval multi-choice accuracy results on MMBench `dev` set. We adopt the following abbreviations: LR for Logical Reasoning; AR for Attribute Reasoning; RR for Relation Reasoning; FP-C for Fine-grained Perception (Cross Instance); FP-S for Fine-grained Perception (Single Instance); CP for Coarse Perception. Baseline results are taken from Liu et al. [123].

LLM	**Data**	**Overall**	**LR**	**AR**	**RR**	**FP-S**	**FP-C**	**CP**
OpenFlamingo$_{9B}$	-	6.6	4.2	15.4	0.9	8.1	1.4	5.0
MiniGPT-4$_{7B}$	5k	24.3	7.5	31.3	4.3	30.3	9.0	35.6
LLaMA-Adapter$_{7B}$	52k	41.2	11.7	35.3	29.6	47.5	38.6	56.4
Otter-I$_{9B}$	2.8M	51.4	32.5	56.7	53.9	46.8	38.6	65.4
Shikra$_{7B}$	5.5M	58.8	25.8	56.7	**58.3**	57.2	**57.9**	**75.8**
Kosmos-2	14M	59.2	**46.7**	55.7	43.5	64.3	49.0	72.5
InstructBLIP$_{7B}$	1.2M	36.0	14.2	46.3	22.6	37.0	21.4	49.0
IDEFICS$_{9B}$	1M	48.2	20.8	54.2	33.0	47.8	36.6	67.1
IDEFICS$_{80B}$	1M	54.6	29.0	**67.8**	46.5	56.0	48.0	61.9
InstructBLIP$_{13B}$	1.2M	44.0	19.1	54.2	34.8	47.8	24.8	56.4
LLaVA$_{7B}$	158k	38.7	16.7	48.3	30.4	45.5	32.4	40.6
LLaVA-SFT^{+}$_{7B}$	220k	52.1	28.3	63.2	37.4	53.2	35.9	66.8
LLaVA-RLHF$_{7B}$	280k	51.4	24.2	63.2	39.1	50.2	40.0	66.1
LLaVA$_{13B\times336}$	158k	47.5	23.3	59.7	31.3	41.4	38.6	65.8
LLaVA-SFT^{+}$_{13B\times336}$	220k	57.5	25.8	65.7	54.8	57.9	51.0	68.5
LLaVA-RLHF$_{13B\times336}$	280k	**60.1**	29.2	67.2	56.5	**60.9**	53.8	71.5

5.5.3 Ablation Analysis

We conduct ablation studies on LLaVA$_{7B}$ and evaluate over the four aforementioned benchmarks. We compare the performance of Fact-Augmented RLHF (Fact-RLHF) with standard RLHF in Table 5.5. Our findings indicate that while the conventional RLHF exhibits improvement on LLaVA-Bench, it underperforms on MMHAL-BENCH. This can be attributed to the model's tendency, during PPO, to manipulate the naive RLHF reward model by producing lengthier responses rather than ones that are less prone to hallucinations. On the other hand, our Fact-RLHF demonstrates enhancements on both LLaVA-Bench and MMHAL-BENCH. This suggests that Fact-RLHF not only better aligns with human preferences but also effectively minimizes hallucinated outputs. [5]

[5]See detailed discussion of ablations on high-quality instruction data in Appendix, and data filtering v.s. RLHF in Appendix

Table 5.5: Abalation studies on methodologies (SFT, RLHF, and Fact-RLHF), data mixtures (LLaVa with additional datasets), and model sizes of the policy model (PM) and the reward model (RM).

Method	PM	RM	SFT Data			MMBench	POPE	LLaVA-B	MMHal-B
			VQA	AOK	Flickr				
SFT	7b	-	✗	✗	✗	38.7	76.0	81.0	1.3
SFT	7b	-	✓	✗	✗	42.9	82.0	30.4	2.0
SFT	7b	-	✗	✓	✗	48.5	79.8	34.7	1.1
SFT	7b	-	✗	✗	✓	37.8	77.6	46.6	1.5
SFT	7b	-	✓	✓	✓	**52.1**	**82.7**	86.3	1.8
RLHF	7b	7b	✗	✗	✗	40.0	78.2	85.4	1.4
RLHF	7b	7b	✓	✓	✓	50.8	**82.7**	87.8	1.8
RLHF	7b	13b	✓	✓	✓	48.9	**82.7**	93.4	1.8
Fact-RLHF	7b	13b	✓	✓	✓	51.4	81.5	**94.1**	**2.1**

5.6 Discussions & Limitations

Hallucination phenomena are observed in both LLMs and LMMs. The potential reasons are two-fold. Firstly, a salient factor contributing to this issue is the low quality of instruction tuning data for current LMMs, as they are typically synthesized by more powerful LLMs such as GPT-4. We expect our proposed high-quality vision instruction-tuning data and future efforts on manually curating high-quality visual instruction tuning data can alleviate this problem.

Secondly, the adoption of behavior cloning training in instruction-tuned LMMs emerges as another fundamental cause [168]. Since the instruction data labelers lack insight into the LMM's visual perception of an image, such training inadvertently conditions LMMs to speculate on uncertain content. To circumvent this pitfall, the implementation of reinforcement learning-based training provides a promising avenue, guiding the model to articulate uncertainties more effectively [89, 117]. Our work demonstrates a pioneering effort in this direction. Figure in appendix illustrates the two sources of hallucination in current behavior cloning training of LLMs.

However, while LLaVA-RLHF enhances human alignment, reduces hallucination, and encourages truthfulness and calibration, applying RLHF can inadvertently dampen the performance of small-sized LMMs. Balancing alignment enhancements without compromising the capability of LMM and LLM is still an unresolved challenge. Though we've demonstrated the effective use of linear projection in LLaVA with top-tier instruction data, determining an optimal mixture and scaling it to bigger models remains intricate. Our research primarily delves into the fine-tuning phase of VLMs, leaving the issues of misalignment in other modalities and during pre-training yet to be explored.

Finally, while MMHal-Bench focuses on curtailing hallucinations when evaluating LMMs, it is noteworthy that short or evasive responses can inadvertently attain high scores on MMHal-Bench. This underlines an intrinsic trade-off between honesty and helpfulness [16]. Consequently, for a more comprehensive assessment of alignment with human preferences, we advocate for the evaluation of prospective LMMs using both MMHal-Bench and LLaVA-Bench.

5.7 Conclusion

We proposed several strategies to tackle the multimodal misalignment problems, particularly for LMM, which often produce text inconsistent with the associated images. First, we enrich GPT-4 generated vision instruction tuning data from LLaVA with existing human-authored image-text pairs. Next, we adopt the Reinforcement Learning from Human Feedback (RLHF) algorithm from the text domain to bridge vision-language gaps, wherein human evaluators discern and mark the more hallucinated output. We train the LMM to optimize against simulated human preferences. Moreover, we introduce the Factually Augmented RLHF, leveraging additional factual information such as image captions to enhance the reward model, countering reward hacking in RLHF, and boosting model performance. For tangible real-world impact assessment, we have devised MMHal-Bench, an evaluation benchmark targeting the penalization of hallucination. Remarkably, LLaVA-RLHF, being the first LMM trained with RLHF, shows a notable surge in performance across benchmarks. We opensource our code, and data and hope our findings could help the future development of more reliable and human-aligned LLMs and LMMs.

Part III

Aligning Language Models Towards Complex Reasoning

Chapter 6

Easy-to-Hard Generalization: Scalable Alignment Beyond Human Supervision

In the previous chapter, we explored how reinforcement learning and factual augmentation can ground multimodal models in truth, reducing hallucinations and encouraging more reliable responses. But truthfulness alone does not guarantee capability—particularly on tasks that demand rigorous, multi-step reasoning far beyond the reach of human supervision. As models grow more powerful, the central question shifts from "How do we keep them accurate?" to "How do we help them reason correctly when humans can no longer guide every step?" In this chapter, we take a step toward answering that question through the lens of easy-to-hard generalization. Rather than relying on human annotations for the most difficult problems, we show how models can learn from supervision on easier tasks and generalize their reasoning abilities to solve challenges that humans cannot readily supervise—laying the groundwork for scalable alignment in complex domains like mathematics.

6.1 Introduction

Rapid advancements in large language models (LLMs) indicate that in the near future, highly sophisticated AI systems could surpass human capabilities in certain areas, significantly enhancing our capabilities in solving harder problems beyond the levels we can currently solve [140, 142]. Since the current AI alignment methods mostly rely on either supervised fine-tuning (SFT) with human-provided demonstrations [40, 165, 206] or reinforcement learning from human feedback (RLHF) [144, 178, 244], their capabilities would be inherently limited as humans cannot always provide helpful demonstrations or supervision on the hard tasks beyond their

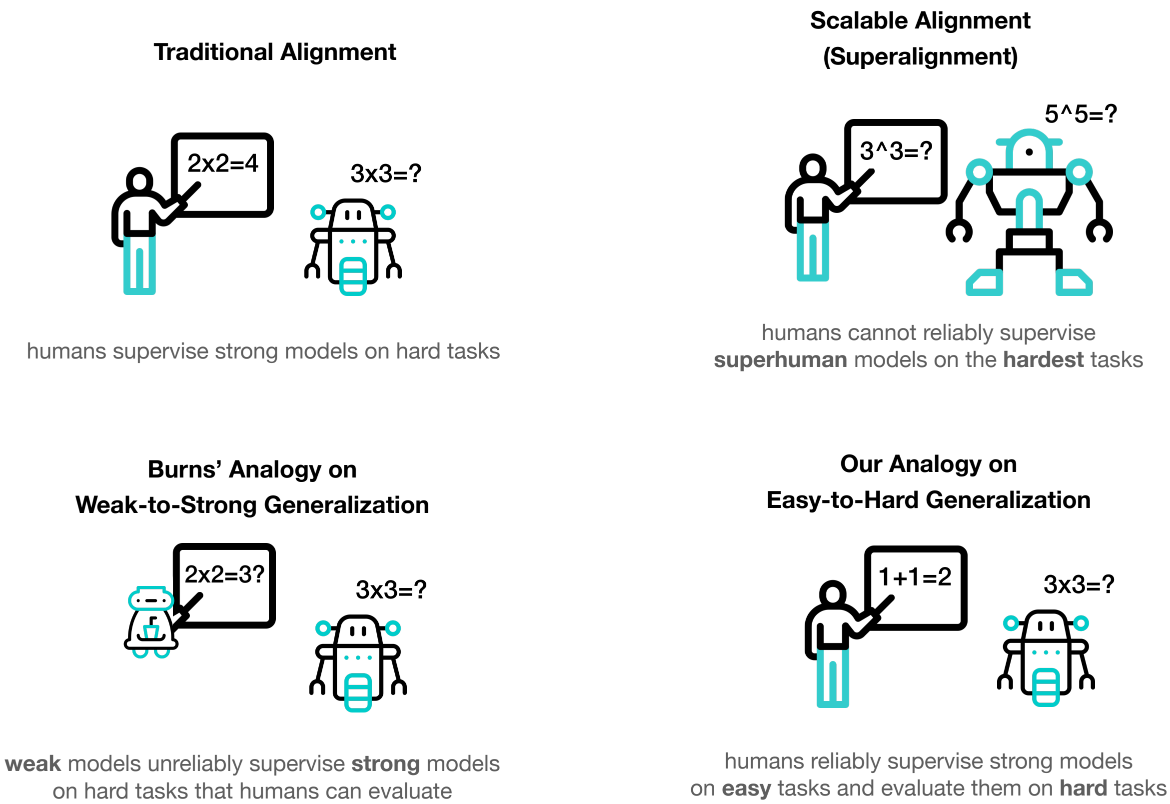


Figure 6.1: Illustration of different alignment scenarios: **traditional alignment** relies on human demonstrations or judgements [144]; **scalable alignment** [22] assumes that humans cannot reliably supervise smarter-than-human models; **weak-to-strong generalization** [24] focuses on using weak models with unreliable labels to supervise strong models; Our proposed **easier-to-general generalization** focuses on the transfer of rewarding policies from weak models to harder tasks.

expertise [172].

In order to build future AI systems for tackling complex challenges, such as advancing scientific knowledge, it is crucial to develop new approaches for *scalable oversight* challenge, i.e., to supervise the AI systems that can potentially outperform humans in most skills [22]. The key question is:

- *Can we limit human supervision to easier tasks, yet enable the model to excel in harder tasks?*

We refer to this scenario as *Easy-to-Hard Generalization* [24, 76, 170, 240]. This setting requires no human supervision on the harder tasks, which differs from existing work that either enhances humans' ability to verify the outputs of AI systems [22, 159, 166, 212] or enables weak-to-strong generalization via a teacher that only offers unreliable or noisy supervision [24].

The most basic form of easy-to-hard generalization can be achieved by training the policy models (i.e., generator) using supervised fine-tuning (SFT) or in-context learning (ICL) on easy tasks [23, 154], and expect this will unlock the ability to perform well on hard tasks. However, it has been observed that SFT or ICL training of generators on easy tasks often fails to generalize to hard tasks [62, 186, 240]. We hypothesize and show that methods beyond these can enable stronger degrees of easy-to-hard generalization. Our intuition is guided by the observation that *evaluation is easier than generation* [92, 134], so an evaluator may offer a degree of easy-to-hard generalization that is useful for improving a generator. If that is true, we can first train a verifier on easy tasks, then make use of its generalization ability to supervise the generator on hard tasks.

Complex tasks can often be broken down into smaller steps [240] and verified by validating the individual steps – a strategy that is commonly employed in solving mathematical problems [114, 195, 196]. Inspired by this, we train outcome-supervised and process-supervised reward models [114, 196, 200, 226] as our easy-to-hard evaluators. The training dataset is often comprised of a set of labeled easy tasks, each with a question and a high-quality solution[1], paired with a set of unlabeled hard tasks that are represented only by their questions. This simulates the practical setting of having numerous problems with known solutions, as well as significant unresolved challenges, such as the Millennium Prize Problems [27], which present challenging open problems. The pivotal aspect of easy-to-hard generalization thus lies in how we effectively leverage the capabilities of easier-level models in solving harder problems.

Our investigation includes to training policy and reward models on the easy (i.e., level 1-3) portion of the PRM800K [114] dataset, and comparing the performance of majority voting with the policy model only and weighted majority voting with the policy model and PRMs (Process-supervised Reward Models). We also introduce the *Outcome & Process Reward Model (OPRM)*, which harnesses the complementary strengths of outcome reward models (ORMs) and process reward models (PRMs): judging if each step in reasoning is correct (like PRMs do) and deciding if the final answer is right (like ORMs do). Our findings reveal a marked performance improvement with the inclusion of reward models, especially on the hard (i.e., level 4-5) portion of the MATH500 test set. This improvement indicates that easier-level evaluators can maintain their effectiveness on harder tasks. We have similar observations in our experiments on the MetaMath dataset [227] and the Math-Shepherd dataset [200].

We further investigate the use of the easy-to-hard evaluator as a reward model in rein-

[1]We assume that human supervision is of high quality on the easy tasks in general.

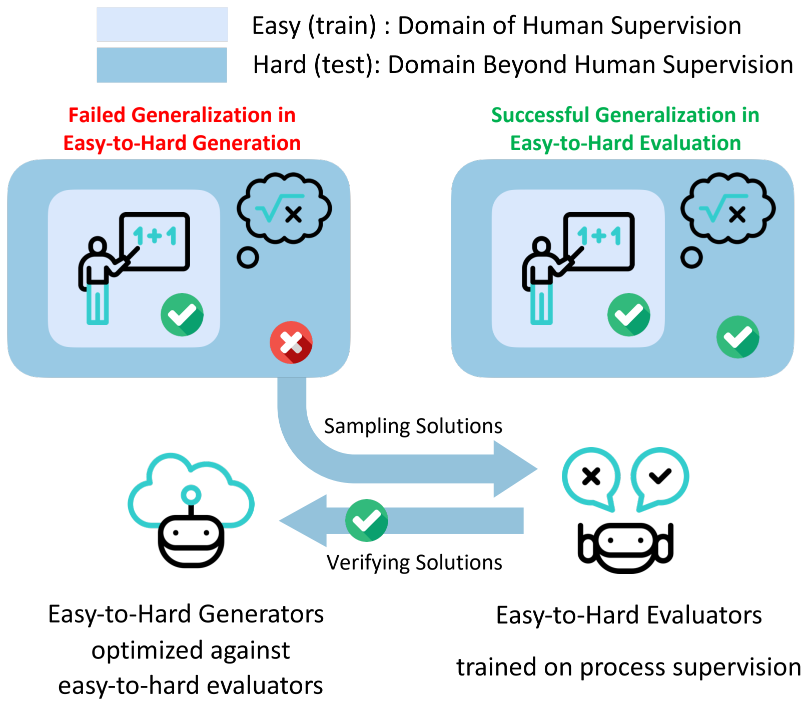


Figure 6.2: We first train the evaluator with process supervision or outcome supervision (which simulates the process supervision) to enable easy-to-hard evaluation, and then use it to facilitate easy-to-hard generation via re-ranking or RL.

forcement learning, where the evaluator provides targeted, step-by-step guidance in solving hard problems. We have an intriguing finding that *training with human supervision only on the easy tasks (i.e., training with Level 1-3 problems and answers) can outperform both SFT and Final-Answer RL training on the full dataset (Level 1-5).* This finding underscores the potential of using easy-to-hard evaluation to improve easy-to-hard generators, particularly when dealing with varied levels of task complexity.

6.2 Related Work

6.2.1 Scalable Oversight

While present-day models operate within the scope of human assessment, future, more advanced models may engage in tasks that are beyond human evaluation capabilities. This raises a concern that such models might prioritize objectives other than maintaining accuracy (Andreas 6, Perez et al. 149, Sharma et al. 172, Wei et al. 208). To address this, a branch of research develops techniques to enhance the human capacity to supervise such models, such as via using AI to evaluate the work of other AIs [5, 22, 105, 166]. Our setting differs from enhancing human oversight; instead, we focus on enabling models to excel in hard tasks where human supervision may not be available. This also differs from weak-to-strong generalization [24], where human supervision may be available, but not reliable, on hard tasks. However, our framework aligns with the "sandwiching" concept proposed for measuring progress in scalable oversight, which involves domain experts evaluating the outputs of AI-assisted non-experts [22, 46, 159].

6.2.2 Compositional Generalization

Compositional generalization is a fundamental aspect of how language works [35]. It refers to the ability to understand and utilize novel combinations based on the understanding of basic concepts and a limited number of their combinations [61]. Recently, least-to-most prompting [55, 240] teaches language models how to solve a complex problem by reducing it to a series of easier sub-problems, achieving easy-to-hard generalization on semantic parsing tasks like SCAN [100] and CFQ [93] with perfect generalization accuracy. In addition, least-to-most prompting has also been successful in mathematical reasoning tasks, specifically in datasets like GSM8K [43] and DROP [56], by teaching language models to solve problems more difficult than those seen in the prompts. This success not only underscores the capacity of language models to effectively break down complex tasks into simpler sub-tasks Perez et al. [147], but also demonstrates their generalization capability in solving these sub-problems.

6.2.3 Easy-to-Hard Generalization

Past work has evaluated easy-to-hard generalization by training easy-to-hard generators on easy tasks using supervised finetune-tuning (SFT) or in-context learning (ICL) [23, 154]. Nevertheless, Swayamdipta et al. [186] showed that the BERT model performs poorly on common-sense reasoning when only trained on easy data. Fu et al. [62] showed similar results for ICL on reasoning tasks like GSM8K [44]. In concurrent work, Hase et al. [76] evaluate the performance of easy-to-hard generators on more datasets and models, and find that ICL or SFT on easy tasks is a strong baseline for multiple-choice tasks like ARC [42] and MMLU [77]. In contrast, we evaluate the easy-to-hard generation performance on the more challenging MATH dataset [79], and show that easy-to-hard evaluation can improve a generator's easy-to-hard generalization beyond ICL and SFT. Iterative machine teaching [122] gives theoretical justification to show that training classifiers from easy to hard examples yield better generalization.

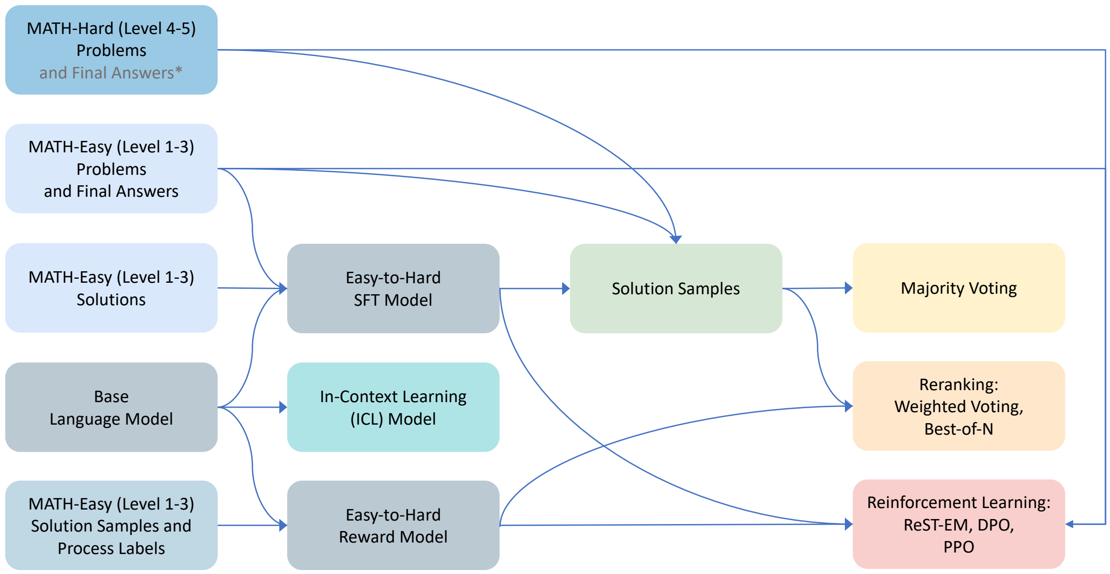


Figure 6.3: The overview diagram of our methods: the different components of modeling and training and how they are interconnected.

6.3 Methodology

We study the easy-to-hard generalization problem: how can we enable capabilities beyond human supervision? Specifically, we explore the efficacy and scalability of various easy-to-hard methodologies on competition-level mathematical problem-solving problems (MATH; Hendrycks et al. 79). This dataset is suitable for our study since it explicitly categorizes problems across five difficulty levels. We consider levels 1-3 as "easy" tasks, encompassing both the problems and their respective solution demonstrations, along with the correct answers. Conversely, levels 4-5, characterized by their more complex nature, are treated as "hard" tasks and are represented solely by their questions. The MATH dataset's difficulty distribution roughly follows a $1:2:2:3:3$ ratio across levels 1 to 5. So our division maintains a balanced number of easy and hard tasks.

The remainder of the paper aims to answer following research questions:

RQ1: How do generators generalize from easy to hard?

RQ2: How do evaluators generalize from easy to hard?

RQ3: If evaluators generalize better than generators, how can we take advantage of this to enable stronger easy-to-hard generalization in generators?

6.3.1 Setup

Dataset MATH [79] is a dataset of 12,500 challenging competition mathematics problems, where 7,500 of them are training problems and 5,000 are originally used for testing. Following Lightman et al. [114], Wang et al. [200], we use the identical subset of 500 representative problems (i.e., MATH500) as our test set, uniformly sample another 500 problems for validation, across all five difficulty levels, and leave the rest 4,000 MATH test split problems combined with the original 7,500 MATH training split problems as our training set.

Simulated Human Demonstrations While the original MATH dataset provides full step-by-step solutions, these solutions typically skip many chain-of-thought steps [207], which can be hard for language models to directly imitate[2]. Instead, we consider filtered PRM800K [114] and MetaMATH [227] as our SFT training data: the former is generated by a Minerva-style base GPT-4 model using few-shot prompting after filtering the correct answers [107, 141], while the latter is generated by ChatGPT [140]. We keep all the GSM8K data in the MetaMATH dataset since they are typically easier than the problems in MATH. PRM800K comes with human annotated process labels, while for MetaMath, we use Math-Shepherd as the corresponding process labels [200].

6.3.2 Generators

For a given dataset (e.g., a variant of MATH), we consider the following generator models:

Full & Hard ICL Full in-context learning (ICL) is a base model prompted with exemplars sampled from all difficulty levels, or only from the level 5 [62].

Easy-to-Hard ICL This model is prompted with exemplars from easy problems. This baseline evaluates the degree to which a model can solve problems more difficult than those seen in the prompts [240].

Full SFT As prior work suggests that finetuning should outperform prompting alone [144, 148, 178], the full supervised fine-tuning (SFT) model is typically considered as a ceiling that a model can achieve on a type of task [24, 76].

Easy-to-Hard SFT This generator model is trained only on the easy tasks. Prior work suggests that it can generalize to hard tasks but with some degeneration in performance [186].

The generator models are evaluated in greedy decoding and self-consistency (also known

[2]Hendrycks et al. [79] found that having models generate MATH-style step-by-step solutions before producing an answer actually decreased accuracy.

as majority voting) settings [203].

6.3.3 Evaluators

Similarly, we consider the following evaluator models that can be trained either on the easy tasks only, or on the full dataset. Notably, unlike final-answer rewards, reward models trained on easy tasks can be applied to evaluate solutions to hard problems.

Final-Answer Reward is a symbolic reward that provides a binary reward based on the accuracy of the model's final answer. The matching is performed after normalization[3].

Outcome Reward Model (ORM) is trained on the Final-Answer rewards. Following Cobbe et al. [43], Lightman et al. [114], Uesato et al. [196], we train the reward head to predict on every token whether the solution is correct, in a similar sense to a value model [226]. At inference time, we use the ORM's prediction at the final token as the reward of the solution.

Process Reward Model (PRM) is trained to predict whether each step (delimited by new-lines) in the chain-of-thought reasoning path is correct. The labels are usually labeled by humans [114, 196] or estimated with rollouts [174, 200].

Outcome & Process Reward Model (OPRM) Building on the distinct advantages of ORMs and PRMs, we introduce the *Outcome & Process Reward Model (OPRM)*, which harnesses the complementary strengths of both. OPRM is trained on the mixed data of ORMs and PRMs. Specifically, it evaluates the correctness of each intermediate reasoning step, akin to PRMs, while also assesses the overall solution's accuracy at the final answer stage, mirroring the functionality of ORMs.

6.3.4 Optimizing Generators Against Evaluators

Finally, given a generator model (i.e., policy model) and a evaluator model (i.e., reward model; RM), we optimize the generator against the evaluator using either re-ranking or reinforcement learning.

Best-of-n (BoN), also known as rejection sampling, is a reranking approach that sample multiple solutions from the generator and selects one with the highest RM score.

Weighted Voting is similar to majority voting or self-consistency [203], but weights each solution according to its RM score [196].

[3] https://github.com/openai/prm800k/blob/main/prm800k/grading/grader.py

Table 6.1: Easy-to-hard generalization of generators. We compare generator performance under various decoding settings. PRM800K and MetaMath indicate the SFT training data and ICL exemplars. Evaluations are performed on the same MATH500 test set.

		PRM800K			MetaMath		
		Greedy	Maj@16	Maj@256	Greedy	Maj@16	Maj@256
Llemma-7b	Full ICL	12.8	15.6	20.8	16.4	18.4	25.6
	Hard ICL	12.6	18.0	27.0	16.6	19.0	27.0
	Easy-to-Hard ICL	14.0	17.6	24.4	14.2	17.4	26.8
	Full SFT	20.6	32.0	36.2	31.4	40.2	41.6
	Easy-to-Hard SFT	19.8	31.6	36.0	30.0	38.6	42.4
Llemma-34b	Full ICL	18.6	23.6	36.0	20.6	28.8	39.2
	Hard ICL	15.8	21.4	34.2	21.8	26.4	38.6
	Easy-to-Hard ICL	18.2	25.2	36.8	19.8	26.8	37.2
	Full SFT	25.6	41.8	46.4	35.4	44.2	45.6
	Easy-to-Hard SFT	24.8	40.8	46.0	32.2	42.6	43.4

Reinforcement Learning (RL) We consider three online/offline RL variants, Reinforced Self-Training (ReST) [72, 175], Direct Policy Optimization (DPO) [158], and Proximal Policy Optimization (PPO) [169]. Due to the space limit, please find their detailed description in Appendix.

6.3.5 Evaluation Metrics

In this study, we have chosen not to establish terms analogous to the weak-to-strong performance gap recovery (PGR) as discussed in Burns et al. [24] or the easy-to-hard supervision gap recovery (SGR) highlighted by Hase et al. [76]. This decision is based on our observations that sometimes, models trained exclusively on simpler tasks—particularly when employing RL training—can outperform those trained across the entire spectrum of problem difficulties. Therefore, we mainly focus on the absolute and relative performance of generators (optionally optimized by the evaluator) on the MATH500 test set [114].

6.3.6 Implementation Details

Base Language Model Llemma is a large language model for mathematics [15], which is continue pre-trained from Code Llama [164] / LlaMA-2 [194]. We use both 7b and 34b variants in our experiments.

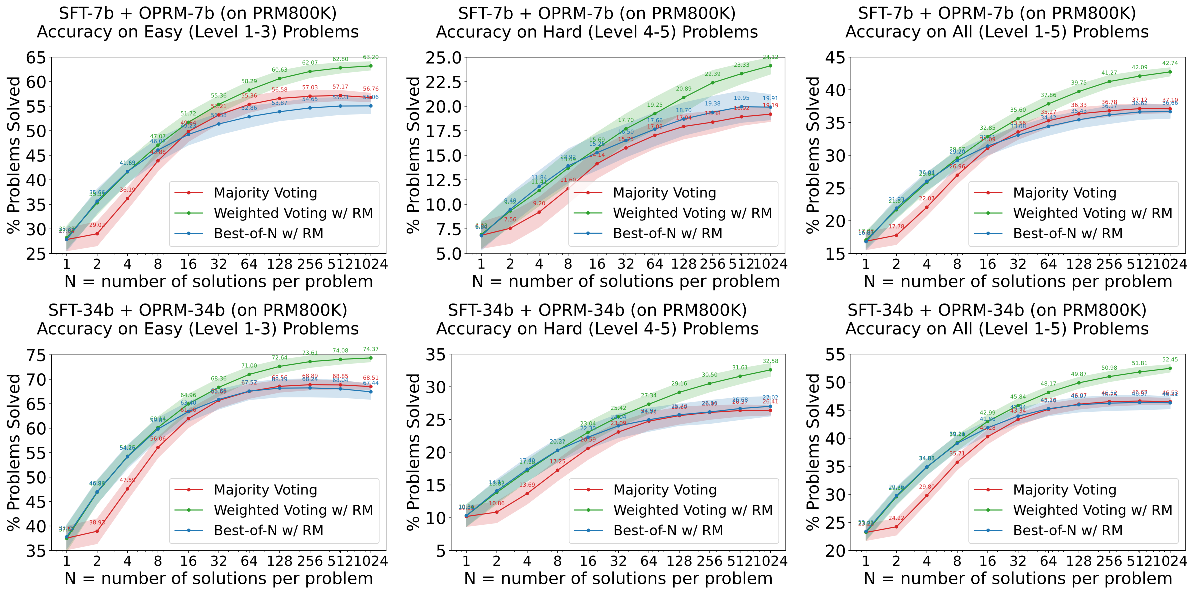


Figure 6.4: Easy-to-hard generalization of 7b (upper) and 34b (lower) evaluators. Both SFTs and RMs are trained on the easy data. We found that PRMs trained on easy tasks can significantly improve the re-ranking (i.e., weighted voting) performance on hard tasks. The shaded margin of the curve plot in this chapter represents the performance variance.

SFT / RL / Reward Model We fine-tune all models in full fine-tuning with frozen input-output embedding layers and normalization layers. RMs are initialized from the base model, and have an added scalar head to output the reward. In PPO training, we initialize the value model from the reward model.

Hyper-parameters Due to the space limit, our training hyper-parameters can be found in Appendix.

6.4 Main Results

6.4.1 Easy-to-Hard Generalization of Generators

In Table 6.1, we compare the easy-to-hard generalization performance of the generators under various decoding settings:

Supervised Fine-Tuning (SFT) outperforms In-Context Learning (ICL): This is consistent with prior work [144, 178, 196]. We also find that the performance of ICL has larger variance than SFT with respect to data ordering (or random seeds) [54, 233].

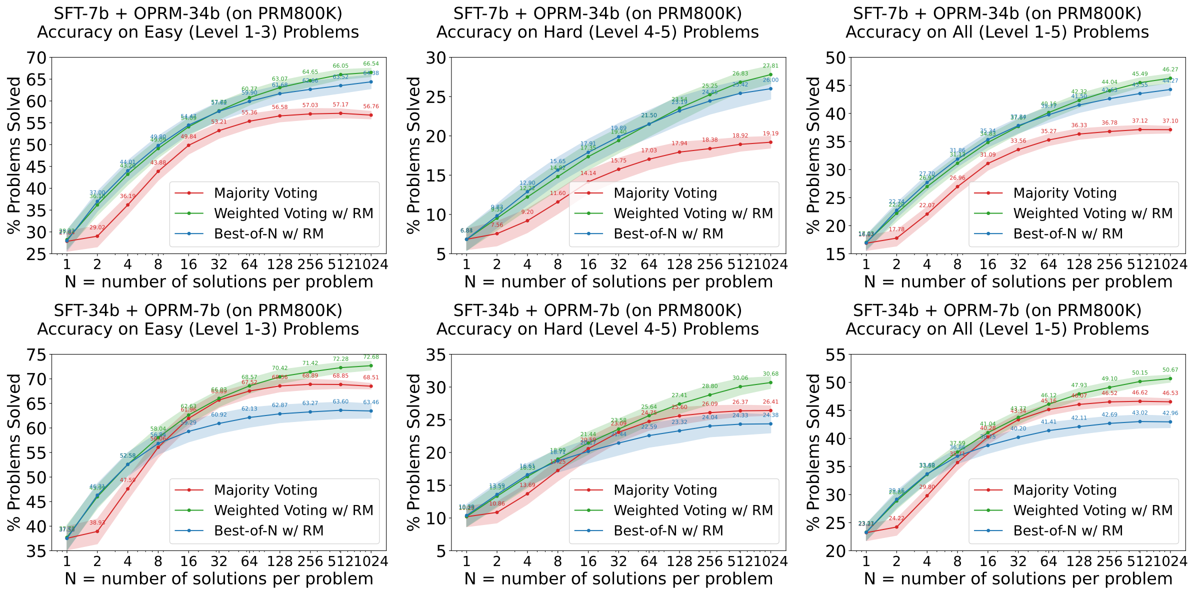


Figure 6.5: Easy-to-hard generalization of evaluators applied to generators of different sizes. We evaluated 7b generator + 34b evaluator (upper) and 34b generator + 7b evaluator (lower). Both SFTs and RMs are trained on the easy data.

SFT data quality impacts easy-to-hard generalization: PRM800K data is generated by a base (unaligned) GPT-4 model through few-shot prompting and is thus of lower quality than well-aligned ChatGPT-generated MetaMATH data. We find that only MetaMath-trained models have certain easy-to-hard gaps (e.g., 16.6 v.s. 14.2 in MetaMath-7b-ICL), while such gaps in PRM800K-trained models are very small (less than 1%), or even inverted in the ICL setting. We hypothesize that low-quality SFT data may only teach the model the format of the task [165, 203, 206], while high-quality (imitation) SFT data can teach the model the principles of solving the task [71, 182]. Nevertheless, the strongest performance is achieved by full SFT on the high-quality MetaMath data (35.4), showing an unignorable difference, with a gap of up to 3.2, compared to its easy-to-hard SFT counterpart (32.2).

6.4.2 Easy-to-Hard Generalization of Evaluators

The primary metric we use to assess the effectiveness of our process reward model is not the average accuracy of verifying each step in a solution but rather the overall performance achieved through re-ranking methods (See discussion in Sec. 6.3.5). We first use re-ranking to evaluate the easy-to-hard generalization performance of evaluators.

Re-ranking

We consider two re-ranking strategies: Best-of-n (or rejection sampling) and Weighted Voting. In our easy-to-hard generalization setting, both SFT models and Reward Models (RMs) are trained on easier tasks (levels 1-3), but evaluated on all difficulty levels (1-5). We compare the performance between majority voting (SFT only) and re-ranking (SFT + OPRM) on the PRM800K dataset in Figure 6.4-6.5, and the performance of different reward models (PRMs, ORMs, & OPRMs) on the PRM800K dataset in Figures in appendix. Specifically, we use `min` as the reward aggregation function for best-of-n and `prod` for weighted voting[4]. The figures illustrate the performance of different decoding strategies or reward models under the same number of sampled solutions per problem. We have the following findings:

OPRMs outperforms ORMs and PRMs This confirms our hypothesis that Process Reward Models (PRMs) and Outcome Reward Models (ORMs) capture different aspects of task-solving processes. By integrating the strengths of both PRMs and ORMs, Outcome & Process Reward Models (OPRMs) demonstrate superior performance. However, follow-up experiments conducted on the MetaMath/Math-Shepherd datasets do not demonstrate significant improvements from incorporating additional ORM training examples. This lack of enhancement may be attributed to the fact that Math-Shepherd is already generated from final-answer rewards. This suggests that there remains a substantial difference between process rewards labeled by humans (e.g., PRM800K) and those generated automatically (e.g., Math-Shepherd).

Weighted voting outshines Best-of-n This finding diverges from past research where minimal performance differences were observed between weighted voting and Best-of-n [114, 196]. Our hypothesis is that this discrepancy arises from our specific experiment, which involves training a less powerful base model (Llemma; Azerbayev et al. 15) on more difficult tasks (MATH; Hendrycks et al. 79). This setup might diminish the effectiveness of the reward model, potentially leading to an over-optimization of rewards [66]. Given these insights, weighted voting is preferred as the primary re-ranking method for further discussions. Nevertheless, Best-of-n still achieves competitive performance to majority voting when producing only one full solution. In Figure 6.5, we also find that the 34b evaluator can significantly improve the 7b generator, while the 7b evaluator can still improve the performance of the 34b generator.

[4] See more detailed analysis of reward aggregation functions in Appendix.

Table 6.2: Comparing reinforcement learning (RL) approaches for easy-to-hard generalization. All methods are of 7b size and evaluated with greedy decoding.

	RL Data	Reward		Accuracy		
		Final-Answer	Process RM	Easy (level 1-3)	Hard (level 4-5)	All
(SFT / PRM trained on level 1-3 of PRM800K)						
SFT				28.2	12.2	19.8
ReST-EM	Easy	Easy	×	33.2	12.6	22.4
Iterative DPO	Easy	Easy	√	42.0	12.2	26.4
PPO	Easy	Easy	×	42.0	14.1	27.4
PPO	All	Easy	√	**45.4**	**14.9**	**29.4**
(SFT / PRM trained on level 1-5 of MetaMath / Math-Shepherd)						
Llemma-based SFT SoTA (Ours)				51.7	13.7	31.4
Previous RL SoTA [200]				-	-	33.0
(SFT / PRM trained on level 1-3 of MetaMath / Math-Shepherd)						
SFT				44.1	14.9	28.8
ReST-EM	Easy	Easy	×	50.4	14.5	31.6
Iterative DPO	Easy	Easy	√	**53.8**	**16.0**	**34.0**
Iterative DPO	All	Easy	√	49.6	10.7	29.2
PPO	Easy	Easy	×	50.8	15.3	32.2
PPO	All	Easy	√	**53.8**	**16.0**	**34.0**

Greater effectiveness of re-ranking on harder tasks: Weighted voting not only consistently surpasses majority voting but also shows a more pronounced advantage on harder tasks. This observation leads to the conclusion that *evaluators demonstrate better easy-to-hard generalization capabilities in comparison to generators.* This motivates us to explore RL approaches that optimize the generator against the evaluator to further improve the performance of easy-to-hard generation.

Reinforcement Learning (RL)

Given the conclusion above, an important question arises: how can evaluators once again assist generators in achieving enhanced easy-to-hard generalization capabilities? We further investigate the enhancement of policy models through RL, utilizing easy-to-hard evaluators as reward models. Similar to re-ranking, SFT and PRM are only trained on easy data. For a fair comparison between PRM800K and MetaMath, we only use vanilla PRMs in the RL training. All the RL methods use the validation accuracy for selecting the best checkpoint[5]. Our comparison spans

[5] This includes stopping iterations in ReST-EM and iterative DPO, and stopping online steps in PPO.

offline (ReST & DPO) and online (PPO) RL algorithms under two training conditions:

Easy Questions & Easy Final Answers. The SFT model samples from easy questions and receives the corresponding Final-Answer and optional PRM rewards.

All Questions & Easy Final Answers. This assumes access to a range of easy and hard problems for RL training, with rewards for hard tasks solely provided by the easy-to-hard evaluator.

Based on the results reported in Table 6.2, we have the following findings:

DPO and PPO excel over ReST. Among the RL algorithms trained on the PRM800K dataset, PPO emerges as the most effective, significantly surpassing both ReST and DPO. On the MetaMATH dataset, PPO and DPO achieve top performance, while ReST shows only marginal improvements over the SFT baseline. The comparative analysis between DPO and PPO across the PRM800K and MetaMATH datasets indicates that while DPO's efficacy is on par with PPO given a high-quality SFT model as initialization, PPO's effectiveness is less contingent on the quality of the underlying SFT model [144, 158].

PRM rewards are more beneficial than Final-Answer rewards for hard tasks. Notably, models trained with PRM rewards with human supervision on the easy tasks (achieving a top performance of 34.0) outperform the previous state-of-the-art model trained across all task levels (33.0). This highlights the effectiveness of leveraging easy-to-hard evaluations to improve generator performance across varying task difficulties.

6.4.3 Easy-to-Hard Generalization on the Coding Domain

We conduct further experiments in the coding domain with the APPS dataset [78]. Similarly to Lightman et al. [114], we sub-sampled 500 questions from the original test set of APPS as our test set. Specifically, we sub-sampled 100 Introductory questions, 300 Interview questions, and 100 Competition questions, following the original distribution in the test set.

In Table 6.3, we compare the performance of SFT-trained Code Llama [164] (7b & 34b) with greedy decoding and best-of-N approach. In the latter, an Outcome Reward Model (ORM) of the same model size is trained to select the best coding one from N sampled solutions.

We found that while the reward model is only trained on the outcome supervision of easy (Introductory) data, it significantly improves the model performance on hard (Interview & Competition) data. These findings extend the premise of easy-to-hard generalization beyond the

Table 6.3: Easy-to-hard generalization of evaluators on coding problems (APPS). Both SFTs and RMs are trained on the easy (Introductory) data. We found that ORMs trained on easy tasks can improve the re-ranking (Best-of-N) performance on hard (Interview & Competition) coding problems.

	SFT / ORM Train Data	Decoding	Average Accuracy (%) Intro.	Inter.	Comp.	All	Strict Accuracy (%) Intro.	Inter.	Comp.	All
Code Llama - 7b	All	Greedy	**31.4**	15.5	**12.2**	**18.0**	17.0	2.3	2.0	5.2
	Easy	Greedy	26.8	14.1	9.5	15.7	11.0	3.0	0.0	4.0
	Easy	Best-of-1	25.4	12.0	0.1	13.5	16.0	2.7	0.0	4.8
	Easy	Best-of-4	27.1	13.8	8.1	15.3	14.0	4.0	0.0	5.2
	Easy	Best-of-16	29.7	**16.3**	11.3	**18.0**	**19.0**	**5.0**	**3.0**	**7.4**
Code Llama - 34b	All	Greedy	37.6	19.9	**11.3**	21.7	22.0	5.0	2.0	7.8
	Easy	Greedy	33.9	19.4	8.5	20.1	21.0	6.0	1.0	8.0
	Easy	Best-of-1	28.5	14.5	4.4	15.3	21.0	3.3	0.0	6.2
	Easy	Best-of-4	36.3	21.3	10.5	22.1	24.0	8.7	1.0	10.2
	Easy	Best-of-16	**45.9**	**25.8**	10.0	**26.6**	**30.0**	**10.7**	**3.0**	**13.0**

confines of mathematical reasoning, suggesting its applicability across diverse domains.

6.5 Conclusion

Our study advances the field of AI alignment by demonstrating the potential of easy-to-hard generalization, where models trained on simpler tasks can be guided to solve more complex problems without direct human supervision on these harder tasks. Through the use of (process-supervised) reward models for evaluating and enhancing policy models, we show that evaluators can facilitate this form of generalization, outperforming traditional training methods. Our findings highlight the effectiveness of re-ranking strategies and reinforcement learning (RL) in leveraging evaluators for performance gains on difficult tasks. This approach presents a promising direction for developing AI systems capable of surpassing human problem-solving capabilities, suggesting a scalable alignment method that could enable AI to independently advance knowledge in complex domains.

While our study provides valuable insights into easy-to-hard generalization and the potential of process-supervised reward models, there are limitations to consider. These include the focus on specific model sizes and datasets, the domain specificity of reasoning tasks, and the need for further research on the long-term implications and robustness of the method.

Chapter 7

Lean-STaR: Learning to Interleave Thinking and Proving

The previous chapter demonstrated how reward models trained on easy tasks can guide language models to solve much harder problems, even those that exceed human evaluative ability. This "easy-to-hard generalization" approach offers a powerful framework for scalable alignment: instead of requiring humans to supervise every complex task, we can train evaluators to generalize supervision and provide feedback even when direct human oversight is unavailable. In this chapter, we extend this philosophy to one of the most rigorous domains of reasoning: formal theorem proving. Here, correctness is not a matter of preference or probability—it must be provable, step by step, in a formal language. We introduce Lean-STaR, a new method that teaches models to interleave informal reasoning (natural language thoughts) with formal steps, allowing them not only to prove theorems, but also to explain each move in a way that is both human-intuitive and machine-verifiable. This approach further illustrates how alignment and reasoning can go hand-in-hand—by making models think out loud before they act, we open a path to greater transparency, rigor, and generalization in the most demanding problem settings.

7.1 Introduction

Theorem proving is a fundamental aspect of mathematics, and mathematical reasoning is an important part of artificial intelligence [136, 239]. *Formalized mathematics* in particular provides a challenging testbed for assessing mathematical reasoning capabilities. Since theorems and proofs in this setting can be represented in the form of checkable source code, it is easy to evaluate proofs of arbitrary complexity [50]. Automated theorem proving, if successful, can also help discover unknown errors in previous proofs[1], and make it easier to guarantee that new proofs are correct. More broadly, formal mathematics coupled with powerful automation may unlock new forms of education and collaboration, mathematical insights, and applications to verifying critical software [12, 25, 59, 139].

Recently, language models have shown promising progress in formal theorem proving [74, 102, 112, 151, 153, 214, 218]. Existing approaches typically train a model solely based on the proofs in a formal language (code) such as Lean [50], Isabelle [137], or Coq [45]. Our key observation is that such approaches ignore a wealth of *informal* information that may be useful for learning to prove theorems [210, 211]. For instance, the underlying *thought process* prior to each step of a proof is not present in formal source code. Based on this insight, we propose to train a language model that can produce a natural language chain-of-thought ("thought") prior to each step ("tactic") of a formal proof.

We introduce Lean-STaR, a framework for learning to interleave informal thoughts with steps of formal proving. Building on the Self-Taught Reasoner (STaR) framework [228], we enable language models to interleave step-by-step rationales (i.e., thoughts) [138, 207] with formal proving in a two-stage process. In an initial phase, we prompt a sufficiently capable language model, such as GPT-4 [1], and generate retrospective thoughts based on a dataset of human-written proofs, such as Mathlib, the largest collection of human-written proofs in Lean [128]. Subsequently, we fine-tune a thought-augmented tactic predictor [20, 21, 47, 68] that, given a Lean state, can generate a thought and predict the subsequent tactic. In a second phase, we optimize this thought-augmented tactic predictor with the expert iteration algorithm [8, 175], using multi-step success rate in theorem proving as the reward.

Our work presents a new link between informal and formal mathematics, complementary to prior explorations that translate standalone mathematical statements [187, 201, 215] or translate informal proofs into formal proofs [2, 14, 82, 87, 241]. Lean-STaR generates natural language thoughts specifically for each proof step, improving formal proving capabilities by interleaving natural and formal languages.

We instantiate Lean-STaR by generating roughly 50,000 thought-augmented examples from Lean's Mathlib [128], then synthesize an additional 50k examples through two iterations of expert iteration. To the best of our knowledge, this yields the first thought-augmented dataset for theorem proving. After fine-tuning an InternLM2-7b base model [223] on our thought-augmented data, our final Lean-STaR model can solve $\mathbf{34.8}\%$ (pass@32) or $\mathbf{36.1}\%$ (pass@64) of

[1]For example, Terence Tao found a non-trivial error while using Lean to formalize a project [188].

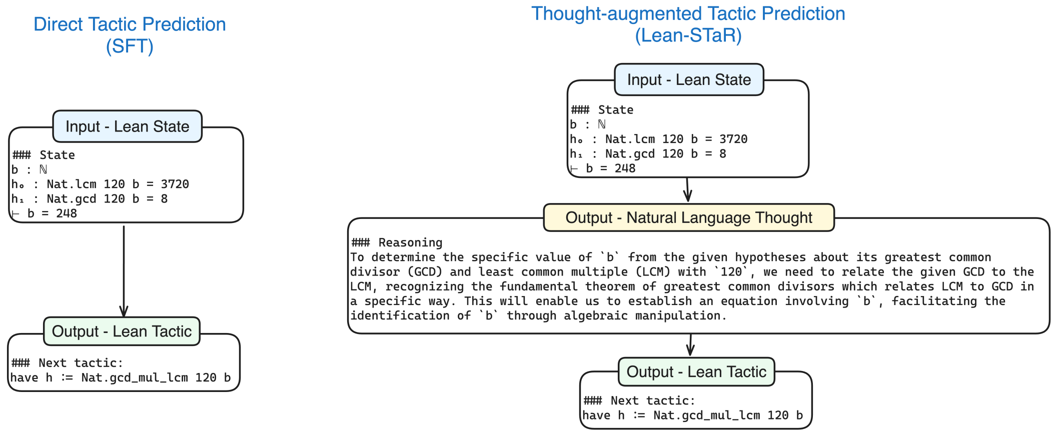


Figure 7.1: The illustration of tactic prediction in one proof step with and without thought.

the problems on miniF2F-test [234]. Using stronger base model InternLM2-7b-plus, Lean-STaR can achieve $\mathbf{45.4\%}$ (pass@32), significantly surpassing the previous results of 43.4% (pass@32). In summary, Lean-STaR offers a framework for teaching language models to interleave informal thoughts with formal verification, advancing the capabilities of language models in automated theorem proving.

7.2 Related Work

Automatic Theorem Proving & Autoformalization. Previous work on learning-based theorem proving typically follows the GPT-f framework [151], which trains a language model on (proof state, next-tactic) pairs, then proves theorems by using the model within a best-first tree search. Subsequent work has explored several directions, including data augmentation [75], novel proof search methods [102, 199], further training through curriculum learning [152], retrieval augmentation [218], or practical tools [209]. Others use prompted models to generate tactics [15, 192], or fine-tune models to generate a full proof [60]. A second *autoformalization* [215] thread incorporates informal mathematics into formal theorem proving. Draft-Sketch-Prove [87] shows that language models have some ability to use informal proofs to improve a model's formal proving abilities, by drafting an informal proof, translating into a formal proof sketch, then completing the proof with tools like Sledgehammer [21]. Draft-Sketch-Prove and related methods [198, 231, 242] are limited to the Isabelle prover, since they use powerful automatic proving tools like Sledgehammer. Lean lacks these tools, so generating the entire proof at once would be more unlikely in Lean. We focus on Lean, and train language

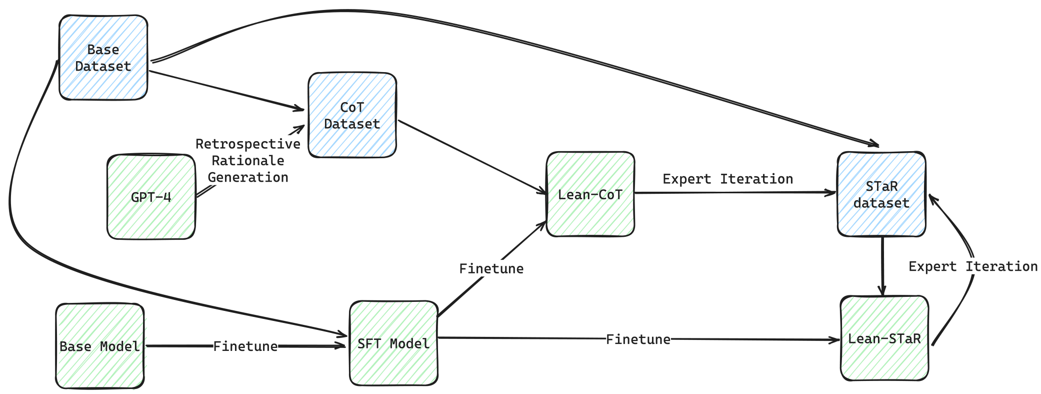


Figure 7.2: **The diagram of our pipeline.** (1) Produce CoT dataset through GPT-4. (2) Fine-tune the SFT model with the CoT dataset to obtain Lean-CoT. (3) Use expert iteration to generate the STaR dataset through the model in the last iteration (Lean-CoT in the first iteration) and then fine-tune Lean-CoT on the updated STaR dataset to obtain the model in the next iteration. We continue performing this step until a stopping condition is met (e.g., a fixed number of iterations).

models to generate a thought and predict the subsequent tactic in each proof step. To the best of our knowledge, we are the first to introduce thought-augmented reasoning in automatic theorem proving.

Rationale-augmented Reasoning. Recently, many works demonstrated that letting language models reason before an answer can improve their performance on tasks including math, science, and code [32, 138, 207]. Although the corresponding techniques (e.g., Scratchpad and Chain-of-Thought) have proven to be effective, they require either extensive annotated training examples or exposure to numerous similar examples during pre-training [23]. The scarcity of natural language reasoning in formal theorem proving, coupled with the impracticality of manually annotating rationales for formal mathematics, thus presents a challenge. We propose a new Lean-STaR framework for *synthesizing* training examples by taking advantage of the correctness signal from the formal system.

Bootstrapping Language Model Reasoning. Recently, several works suggest that language models may be taught to reason via synthetic data that they generate themselves, akin to a reinforcement learning method that improves a policy through self-play. Polu et al. [152] showed that a simple RL algorithm, expert iteration, paired with curriculum learning can improve a formal theorem proving model. Self-Taught Reasoner (STaR) [228] showed that we can itera-

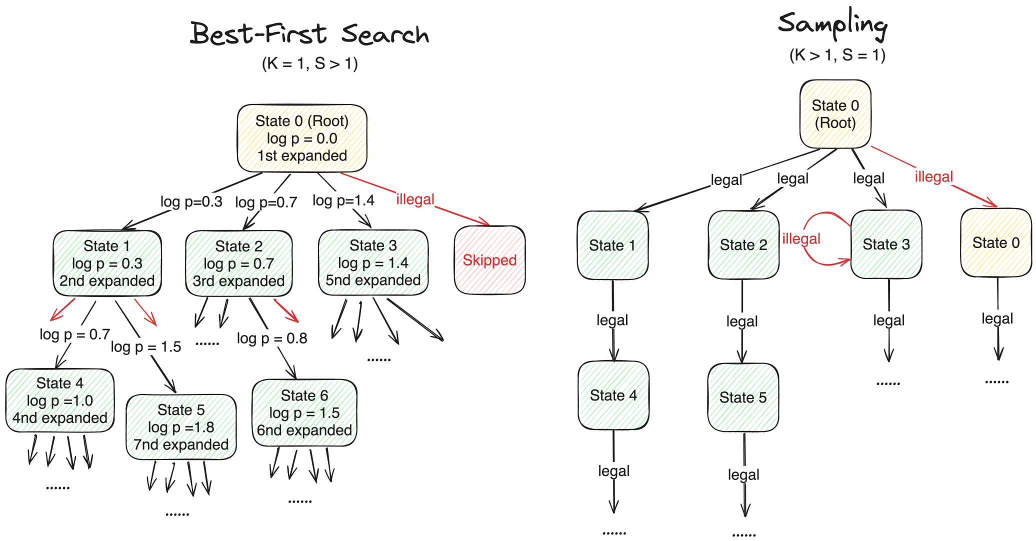


Figure 7.3: **The visualization of Best-first Search ($K = 1$) and Sampling ($S = 1$).** Search method maintains a search tree and explores S tactics on each expanded node. Sampling method explores K tactic trajectories from the root and ignores illegal tactics in the trajectories.

tively fine-tune the language model on the correct (reasoning, answer) pairs generated by itself to gradually improve performance. Singh et al. [175] proposed ReST-EM, which filters data generated by language model with a binary feedback signal rather than using fully manually annotated data (similar to expert iteration in [152]). Our work builds on these ideas, providing the first study of bootstrapped thought-augmented proving.

7.3 Our Method: Lean-STaR

We introduce Lean-STaR, a new method for combining informal thoughts with formal theorem proving. First, we recap interactive theorem proving (§7.3.1). Then we present Lean-STaR's data-generation (§7.3.2, §7.3.2) and reinforcement learning (§7.3.2) phases. Finally, we present our evaluation protocols (§7.3.3).

7.3.1 Preliminaries

Interactive Theorem Provers (ITPs) are typically used for step-by-step automatic theorem proving in formal mathematics. At each step, we can provide the ITP with a high-level "tactic" to simplify the current goal state (e.g., the initial goal theorems to be proven) into subgoals. These

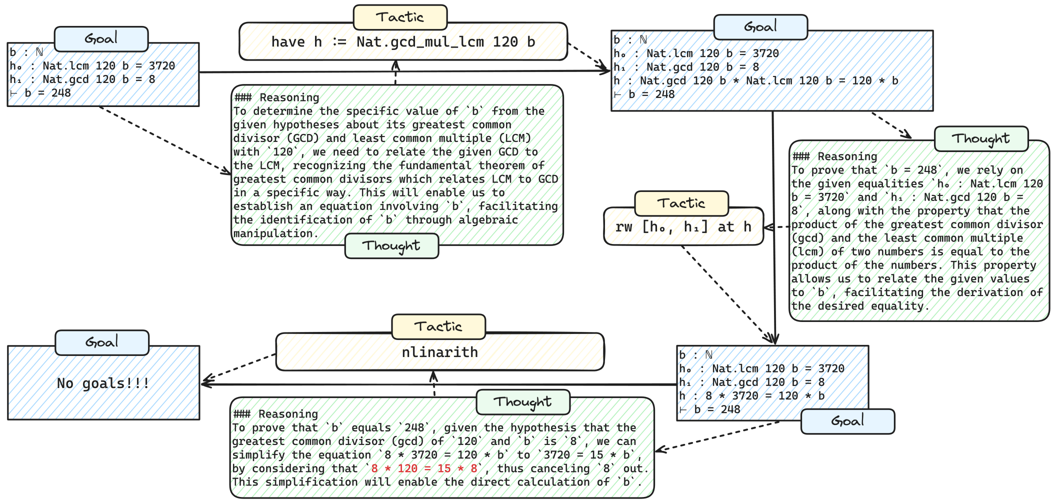


Figure 7.4: **An example of Lean proof and thoughts generated by Lean-STaR**. Note that there is a calculation error in the thought (in red), but this does not affect the correctness of the proof because the calculation task is actually completed by the interactive theorem prover (i.e., Lean's `nlinarith`) instead of the language model. This shows a benefit of combining neural and symbolic systems.

subgoals will form new states, and proving all the subgoals results in a complete proof of the given theorem. We use Lean [50], a popular interactive theorem prover. An example formal proof in Lean and its explanation are shown in Appendix.

7.3.2 Data Generation & Training

We describe the data generation and training of the direct tactic prediction model (SFT), the thought-augmented tactic prediction model trained with synthetic data (Lean-CoT), and the final model trained with expert iteration (Lean-STaR).

Direct Tactic Prediction

We define the theorem-proving problem as a *Markov Decision Process* (MDP) $(\mathcal{S}, \mathcal{A}, P_a, R_a)$ where proof states serve as states in MDP and tactics serve as actions. From this perspective, a proof is a trajectory $(s_1, a_1, r_1), (s_2, a_2, r_2), \cdots$ of states s_i, tactics a_i, and rewards $r_i \in \mathbb{R}$, and the ITP (e.g., Lean) provides each new state s_{i+1}.

In the typical setting [151], proving a theorem consists of providing a proof state s to the language model and then generating a tactic from the language model M, i.e., $\pi_M(a|s)$. The language model can be fine-tuned for this task using a dataset of (proof state, next-tactic) pairs from successful proof trajectories, i.e. $D = \{(s^i, a^i) : i = 1, \cdots, M\}$, where final states have a reward of 1. We refer to a language model fine-tuned on such a dataset as a *supervised fine-tuning (SFT)* model.

Thought-augmented Tactic Prediction

Existing approaches typically train only on formal states and tactics [151]. We hypothesize that incorporating a latent *thought* can improve a model's ability to predict the next tactic. Formally, we introduce a hidden "thought" variable t_i prior to each tactic, and then extend the model to the form $\pi_M(a_i, t_i|s_i) = \pi_M(a_i|t_i, s_i)\pi_M(t_i|s_i)$. In thought-augmented tactic prediction, the distribution over the next tactic can then be expressed as:

$$\pi_M(a_i|s_i) = \sum_{t_i} \pi_M(a_i|t_i, s_i)\pi_M(t_i|s_i).$$

The key challenge is obtaining (state, thought, tactic) pairs for training a model. To this end, we introduce **retrospective rationale generation**. Our motivating observation is that the distribution of natural language thoughts in theorem-proving $\pi_M(t_i|s_i)$ is scarce in the pre-training corpus of large language models. In turn, we find that even the most powerful GPT-4 model does not perform well in generating the correct rationale through few-shot prompting [23]. To develop a language model capable of generating thoughts and tactics $a_i, t_i|s_i$, we need an entirely new dataset $D_T = \{(s^i, t^i, a^i) : i = 1, \cdots, N\}$. However, in Lean, we only have a dataset of $D_S = \{(s^i, a^i) : i = 1, \cdots, N\}$ where (s^i, a^i) is one step in some successful proof trajectories. Given a powerful large language model G, which we refer to as the oracle model[2], we give the oracle model the ground-truth tactic a_i and let the oracle model produce the thought t_i given the current state s_i and ground-truth tactic a_i. This helps improve the pass rate and produce thought-augmented data more efficiently. Our few-shot prompt is provided in Appendix. The design principle of the prompt is to prevent the oracle model from generating hindsight-like thoughts.

We randomly select M pairs $(s^i, a^i) \in D_S$. Then the oracle model is used to produce a thought t^i for each pair (s^i, a^i) to create a new dataset $D_T\{(s^i, t^i, a^i) : i = 1, \cdots, M\}$. With this retrospectively annotated dataset by the oracle model D_T, we obtained our first thought-augmented tactic prediction model, Lean-CoT, by fine-tuning from the SFT model.

[2]For instance, in our experiments we use the best available large language model, GPT-4.

Bootstrapping Thought-augmented Theorem Proving

We propose to apply expert iteration to further improve the performance of Lean-CoT. Specifically, we start from the initial Lean-CoT model M_0 and the initial dataset $D = \{s^i : i = 1, \cdots, M\}$, which consists of all initial states s^i of the theorems to be proved. In iteration 1, we use model M to sample K times per problem. Each time the model will produce a proof trajectory $[(s_0, t_0, a_0), (s_1, t_1, a_1), \cdots, (s_n, t_n, a_n)]$. Then we create a new dataset D_1 by filtering the generated trajectories to include only the successful ones. De-duplication is then applied to the collected trajectories. Now, we can further fine-tune the SFT model M on dataset $D_T \cup D_1$ to produce Lean-STaR model M_1. Then we can use M_1 as initial model to produce dataset D_2 and further fine-tune to obtain model M_2 in the next iteration.

This method can be seen as an offline RL method [175] in the theorem proving MDP. In this MDP, the cumulative reward $R\left((s_0, t_0, a_0), (s_1, t_1, a_1), \cdots, (s_n, t_n, a_n)\right) = 1$ if and only if the proof trajectory is successful. The total expected reward is

$$J(M, D) = \sum_i \mathbb{E}_{(s_0,t_0,a_0),\cdots,(s_n,t_n,a_n)\sim\pi_M(\cdot|s^i)} R\left((s_0, t_0, a_0), \cdots, (s_n, t_n, a_n)\right),$$

and Lean-STaR's expert iteration can be seen as optimizing this reward [175].

7.3.3 Evaluation

Setup. We evaluate the model on formal theorem proving – given a theorem statement, produce a theorem that is correct according to the formal system. This requires an algorithm for producing a full proof by interacting with Lean. As a new form of theorem-proving system, it is unclear what the best strategy is when we have informal thoughts. Our preliminary experiments indicate that best-first search with beam search does not work well for the thoughts in the natural language format. Thus we describe the traditional strategy (best-first search), and our new approach based on sampling.

Best-First Search. The most popular method to evaluate the theorem proving ability of a language model M is to use best-first search like GPT-f [15, 151, 209, 218]. In best-first search, we keep all unexpanded states s_i. Each time, we expand the "best" state s_i and use the language model to sample S next tactics $a_{i,1\cdots S}$ for the current state s_i. For each legal tactic $a_{i,j}$, a new state can be obtained by applying tactic $a_{i,j}$ on state s_i. Following standard practice [151, 209, 218], we assume the state with maximum negative log-probabilities is the "best"s. Specifically, we select state s_i with maximum $\sum_{j=0}^{i-1} -\log p(a_j, s_j)$, where $(s_0, a_0), \cdots, (s_{i-1}, a_{i-1})$ is the proof trajectory before state s_i and $\log p(a_j, s_j)$ is the average log probability of each generated token. We expand up to N states and we get a successful proof search when we reach any proof state with no goals. Then, we can attempt the search K times to obtain a pass rate $pass@K$. However, we found that the best-first search method performed poorly in the Lean-CoT and Lean-STaR models, as detailed in the Appendix. We attribute this to using average log probabilities, which may not be a reliable quality indicator when the thought sequence t_j is generated.

Sampling. Motivated by these issues with applying best-first search to thought-augmented proving, we develop a new method based on sampling trajectories in parallel. Specifically, our method samples K times in parallel for each problem, each time generating at most N tactics. Also, illegal sampled tactics will be ignored during sampling. Specifically, in a sample, suppose our current state is s_i, the proof trajectory before s_i is $(s_0, a_0), \cdots, (s_{i-1}, a_{i-1})$ and the sampled tactic is a_i. If a_i is a legal tactic, (s_i, a_i) will be added to the proof trajectory and we will reach a new state obtained by applying tactic $a_{i,j}$ on state s_i. Otherwise, we ignore this a_i and use language model M to sample a new tactic given state s_i. We limit the number of times a tactic can be generated by language model M to a total of N per time in K sampling times. The sampling method is roughly equivalent to the search with $S = 1$, except that the sampling ignores illegal tactics. We assume that in the sampling method we have $S = 1$. In this setting, evaluating our sampling method and best-first search with equal $S \times K$ took approximately the same amount of GPU time. This sampling method can easily accommodate hidden variable "thoughts" t_j. Figure 7.3 compares best-first search and our sampling method.

7.4 Experiments

We instantiate Lean-STaR using the best available open language model pre-trained on the Lean corpus (InternLM2-Math-base-7b [223]), and follow standard practice in using Lean's Mathlib as the underlying training set (via the Lean Dojo dataset [218]). We generate an initial set of thoughts for Mathlib using GPT-4, perform two rounds of expert iteration, then evaluate the model on miniF2F [234] and leandojo [218], the de-facto standard benchmark for evaluating language-model based theorem provers. Our experimental results show that both retrospective rationale generation and expert iteration significantly improve the theorem-proving capabilities of language models in this setting. We describe our setup and findings in detail below.

Table 7.1: **Pass rates on the minif2f-test and Leandojo dataset with Lean.** This table shows the pass rates of previous works and our work. S is the number of tactics attempted at each expanded node (assumed to be 1 in sampling) and K is the total number of search or sampling attempts per problem. In sampling we use temperature 0.7, and in search we use beam search when generating the next tactic. We use a random subset of Leandojo4-v9-test (novel premises) with a size of 320 as test set of leandojo. Note that we sample 32 examples twice when $K = 64$ in sampling.

Approach	Decoding	N	K	S	minif2f	leandojo
GPT-3.5 [1] (Few-Shot)	Sampling	50	1	1	2.8%	-
GPT-4 [1] (Few-Shot)	Sampling	50	1	1	11.9%	-
Transformer [152] (w/o RL)	Search	512	1	8	24.6%	-
Llemma-34b [15]	Search	50	1	32	25.8%	-
Llemma-7b [15]	Search	50	1	32	26.2%	-
ReProver [218]	Search	50	1	64	26.5%	-
Transformer [152] (w/ RL)	Search	512	1	8	29.6%	-
InternLM2-34b [223]	Search	50	1	32	29.5%	-
COPRA (with GPT-4) [192]	Customized	-	60	1	29.9%	-
COPRA (with GPT-4) [192]	Customized	-	100	1	30.7%	-
InternLM2-7b [223]	Sampling	50	32	1	28.7%	29.7%
InternLM2-7b [223]	Search	50	1	32	30.3%	-
SFT (InternLM2-7b)	Sampling	50	32	1	29.5%	30.6%
SFT (InternLM2-7b)	Search	50	1	32	30.7%	-
Lean-CoT (InternLM2-7b)	Sampling	50	32	1	32.8%	35.6%
Lean-STaR (Iter-1) (InternLM2-7b)	Sampling	50	32	1	34.0%	38.4%
Lean-STaR (Iter-2) (InternLM2-7b)	Sampling	50	32	1	**34.8%**	**39.4%**
Lean-STaR (Iter-2) (InternLM2-7b)	Sampling	50	64	1	**36.1%**	-

7.4.1 Experimental Setup

We use *LeanDojo Benchmark 4 v9* as the supervised fine-tuning (SFT) dataset containing $231,240$ data examples. We fine-tune for 1 epoch to obtain the SFT model. For the learning rate, we use a warmup in the first 20% steps from 0 to 2×10^{-5}, followed by a cosine schedule decaying to zero.

We randomly select $17,256$ different successful proof trajectories from *LeanDojo Benchmark 4 dataset* [218], and use GPT-4-0125 [142] to annotate $52,438$ thoughts from those proof trajectories. We filtered out all proof steps (s^i, a^i) for which a^i contains the newline symbol "\n" before annotating. We perform two iterations of expert iteration, and provide the details in

Table 7.2: **Pass rates about InternLM2-Plus-7B on the minif2f-test dataset with Lean.** This table shows the pass rates of previous works and our work. The evaluation setting is the same as Table 7.1.

Approach	Decoding	N	K	S	Pass rate
InternLM2-plus-7b [223] (from paper)	Search	1000	1	32	43.4%
InternLM2-plus-7b [223] (reproduced)	Search	1000	1	32	42.6%
InternLM2-plus-7b [223]	Sampling	50	32	1	40.9%
SFT (InternLM2-plus-7b) [223]	Sampling	50	32	1	41.3%
Lean-CoT (InternLM2-plus-7b)	Sampling	50	32	1	43.4%
Lean-STaR (Iter-1) (InternLM2-plus-7b)	Sampling	50	32	1	45.4%
InternLM2-plus-7b [223]	Sampling	50	64	1	42.2%
SFT (InternLM2-plus-7b) [223]	Sampling	50	64	1	43.4%
Lean-CoT (InternLM2-plus-7b)	Sampling	50	64	1	45.5%
Lean-STaR (Iter-1) (InternLM2-plus-7b)	Sampling	50	64	1	**46.3%**

Table 7.3: Results for the InternLM2-plus-7b and our Lean-CoT, Lean-STaR, and expert iteration without CoT. We used sampling with $N = 50, K = 32, \& T = 0.7$.

Approach	*Pass@32* of InternLM-Base	*Pass@32* of InternLM-Plus
Few-Shot	28.7%	40.9%
SFT	29.5%(+0.8%)	41.3%(+0.4%)
Lean-CoT	32.8%(+**3.3**%)	43.4%(+**2.1**%)
Lean-STaR	34.0%(+1.2%)	45.5%(+**2.1**%)
Expert iteration (SFT)	30.7%(+1.2%)	43.0%(+1.7%)

Appendix due to space.

We evaluate our method on the *MiniF2F* benchmark [234]. We use a similar evaluation setting as previous works [209, 218, 223], but use our sampling method instead of best-first search for the evaluation of our thought-augmented theorem proving model as discussed in (§7.3.3). We choose these settings to resemble the inference budget used in our baselines, which follow previous work [15, 209, 223]. Namely, for best-first search baselines we use beam search to generate the next tactic with $S = 32, K = 1$ [15, 209, 223]. We do not compare with methods designed for other formal languages such as Jiang et al. [87], Xin et al. [216] since language differences greatly influence the pass rate due to the different tactics and automation. We also do not compare with Lample et al. [102] since they only report $S = 32, K = 64$ on best-first search, which is approximately equivalent to $S = 1, K = 512$ for the sampling method, which is too computationally expensive for us.

7.4.2 Main Results

Our main results are reported in Table 7.1. Lean-STaR gives a significant improvement over the base model. For instance, with a similar inference budget, Lean-STaR achieves 34.8% versus 30.3% in InternLM2 [223] using best-first search and 30.7% in COPRA [192] using GPT-4. With a larger compute budget, Lean-STaR's performance improves further to 36.1%.

Thought augmentation improves theorem proving. The first phase of Lean-STaR trains a model to interleave thoughts and tactics, by fine-tuning on a synthesized dataset of thought-augmented examples. The fine-tuned model from this phase, denoted Lean-CoT in Table 7.1, achieves a pass rate of 32.8%, which is higher than the model prior to this phase, denoted SFT (29.5%). We conclude that the first phase of Lean-STaR can improve the theorem proving ability of a language model, even one that is already specialized for generating tactics in Lean such as the SFT model.

Bootstrapping improves thought-augmented theorem proving. The second phase of Lean-STaR consists of generating new thoughts and tactics with the current language model, saving those that result in correct proofs, and training on the union of the initial thought-augmented dataset and the saved examples (i.e., expert iteration [152, 175, 228]). Refer to Appendix for details.

We perform two iterations of expert iteration, and present the results in Table 7.1, denoted Lean-STaR. Each iteration improves the model's theorem proving performance, from 32.8% (the initial model) to 34% (Lean-STaR after iteration 1) to 34.8% (Lean-STaR after iteration 2). Furthermore, we find that the model is amenable to further improvement via additional sampling, achieving 36.1% by doubling the sampling budget. We conclude that Lean-STaR's second phase can further improve a model's ability to generate thoughts and tactics that lead to correct proofs. We include three qualitative examples in the Appendix, which show the model interleaving thoughts and proof steps.

7.4.3 Experiments with stronger base model and more data

We instantiate Lean-STaR using a stronger language model (InternLM2-Math-plus-7b [223]), which was released after the experiment above. We follow a similar setup to the previous experiment.

In this experiment, we used $140,000$ thoughts annotated by GPT-4o [142] to fine-tune a model ("Lean-CoT"). Then we performed only one iteration of expert iteration and collected about $60,000$ (proof state, thoughts, next-tactic) pairs in data, named "STaR dataset" D_1. We further fine-tuned the Lean-CoT model on dataset D_1 to get the Lean-STaR model.

Our new results are reported in Table 7.2. We can see that Lean-STaR still gives a significant improvement over the baseline. For instance, Lean-STaR achieves 45.4% versus 39.8% in InternLM-plus using sampling with a similar inference budget and 43.4% using best-first search with more inference budget reported in [223]. This results show that both retrospective rationale generation and expert iteration can improve the theorem-proving capabilities on a stronger base model.

7.4.4 Experiments on expert iteration without CoT

Table 7.3 shows the result of expert iteration without CoT (i.e., using (state, tactic) pairs only) as well as the result of Lean-CoT and Lean-STaR. Expert iteration alone achieves 43.0%, which is less than Lean-STaR (45.4%) in InternLM-plus and achieves 30.7% verus 39.8% in InternLM-base. This shows that Lean-STaR's performance gains do not only come from the use of expert iteration.

7.5 Conclusion & Limitations

In this paper, we presented Lean-STaR, a novel approach that significantly enhances the theorem-proving capabilities of language models in formal mathematics by integrating Chain-of-Thought (CoT) rationales into each proof step. Our method begins with generating synthetic rationales using ground-truth tactics retrospectively, followed by fine-tuning the language model to generate these rationales and predict subsequent tactics, resulting in the Lean-CoT model. We further improved this model using expert iteration, fine-tuning it on correct proofs it samples and verifies using the Lean solver. Our contributions include the introduction of the first thought-augmented theorem proving dataset, demonstrating that expert iteration can further improve performance, and achieving new results on the miniF2F-test benchmark, increasing the pass rate from 30.3% to 36.1%. These advancements are not only about improving the accuracy of automated theorem proving, but also offer a scalable and efficient framework for advancing human understanding of mathematics, which may lead to significant impacts in education, scientific discovery, and program verification [12, 28, 59, 90, 139, 187].

The primary limitation of our method is that its performance may be constrained by issues of computational scalability. Both Lean-CoT and Lean-STaR have been fine-tuned on a dataset that is not very large. Additionally, the use of GPT-4 to generate synthetic data may incur a significant cost and possibly introduce biases. Also, expert iteration could face a bottleneck due to CPU and IO limitations, which might slow down the process due to a sluggish speed of Lean ITP.

The success of Lean-STaR in formal mathematics highlights a broader lesson from this thesis: as AI systems take on increasingly complex and high-stakes tasks, alignment must go beyond preference modeling or factual accuracy—it must be integrated into the reasoning process itself. By interleaving informal thoughts with formal proof steps, Lean-STaR exemplifies how models can be taught to reason transparently, verifiably, and in a manner consistent with human intuitions. The final conclusion chapter now reflects on how the combination of scalable alignment strategies—spanning principle-driven learning, fact-grounded optimization, and process-based reasoning—can form the cornerstone of robust and reliable AI systems as we move toward more general capabilities.

Part IV

Conclusion

Chapter 8

Conclusion

This thesis has presented a suite of scalable alignment frameworks that empower large language models to adhere to human values, maintain truthfulness, and perform complex reasoning with minimal human oversight. By introducing a principle-driven alignment strategy and leveraging reinforcement learning from AI feedback, the work demonstrates that models can be guided by a concise set of human-authored principles while reducing the dependency on extensive human annotations. Innovative methods such as recitation augmentation and factually augmented RLHF have been shown to effectively mitigate hallucinations and ground outputs in verifiable information. In addition, the easy-to-hard generalization framework and the Lean-STaR approach have significantly advanced the ability of models to reason through complex tasks and generate transparent, high-quality solutions, thereby establishing a robust foundation for aligning AI behavior as capabilities scale.

Looking ahead, these findings open new avenues for further research into scalable oversight and self-supervised alignment techniques. Future work should focus on developing hybrid systems that integrate automated AI feedback with selective human intervention, enhancing the interpretability and safety of increasingly autonomous systems. Advancements in areas such as adversarial robustness, error detection, and dynamic adjustment of internal reward mechanisms will be critical to ensure that AI systems remain aligned with ethical standards and factual correctness as they tackle more challenging tasks. Continued exploration of grounding mechanisms and improved feedback loops will help refine these alignment strategies, ensuring that future models not only achieve superior performance but also maintain unwavering adherence to human values and societal norms.

These breakthroughs represent pivotal steps on the path toward Artificial General Intelligence, promising a future where AI systems are both extraordinarily capable and inherently aligned with human interests. As scalable alignment frameworks evolve, they pave the way for AGI that can amplify human potential while mitigating the risks associated with superintelligent behavior. The integration of principled alignment with advanced reasoning and self-supervised techniques lays a robust foundation for a transformative era in AI, where the development of safe and beneficial AGI becomes an achievable reality. With sustained research and innovation, the prospect of harnessing AGI for solving complex global challenges and driving unprecedented technological progress is closer than ever before.

Part V

Appendices

8.1 Appendix of Chapter 2

Due to space constraints, we refer readers to the appendix sections in [182] for additional details.

8.2 Appendix of Chapter 3

Due to space constraints, we refer readers to the appendix sections in [181] for additional details.

8.3 Appendix of Chapter 4

Due to space constraints, we refer readers to the appendix sections in [183] for additional details.

8.4 Appendix of Chapter 5

Due to space constraints, we refer readers to the appendix sections in [180] for additional details.

8.5 Appendix of Chapter 6

Due to space constraints, we refer readers to the appendix sections in [184] for additional details.

8.6 Appendix of Chapter 7

Due to space constraints, we refer readers to the appendix sections in [115] for additional details.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7.1, 7.1

[2] Ayush Agrawal, Siddhartha Gadgil, Navin Goyal, Ashvni Narayanan, and Anand Tadipatri. Towards a mathematics formalisation assistant using large language models. *arXiv preprint arXiv:2211.07524*, 2022. 7.1

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*. 5.1

[4] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. 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)*, pages 2357–2367, 2019. 1.4

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016. 3.1, 6.2.1

[6] Jacob Andreas. Language models as agent models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5769–5779, 2022. 3.1, 6.2.1

[7] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 1.3

[8] Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017. 7.1

[9] Anthropic. Core views on ai safety: When, why, what, and how, 2023. URL https://www.anthropic.com/index/core-views-on-ai-safety. 2.1, 2.1, 3.1

[10] Isaac Asimov. Three laws of robotics. *Asimov, I. Runaround*, 2, 1941. 1.2, 3.1

[11] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021. 1, 1.1, 1.2, 4, 2.6, 2.7.2, 3.1, 5.5.2

[12] Jeremy Avigad. Mathematics and the formal turn, 2023. 7.1, 7.5

[13] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 5.1

[14] 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*, 2023. 7.1

[15] 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*, 2023. 6.3.6, 6.4.2, 7.2, 7.3.3, 7.1, 7.4.1

[16] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 1, 1.1, 2.1, 2.7.1, 2.2, 2.7.2, 2.7.3, 3.1, 3.1, 3.2, 3.2, 3.2, 3.3, 3.5.1, 5.1, 5.3, 5.4, 5.5.2, 5.6

[17] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham,

Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. 1, 1.2, 1.2, 2.1, 2, 2.1, 2.4, 6, 2.7.1, 2.7.3, 3.1, 3.1, 3.2, 3.2, 3.1, 3.3, 3.4, 3.4

[18] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023. 1

[19] Ali Furkan Biten, Lluís Gómez, and Dimosthenis Karatzas. Let there be a clock on the beach: Reducing object hallucination in image captioning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390, 2022. 1.3

[20] Jasmin Christian Blanchette, Cezary Kaliszyk, Lawrence C Paulson, and Josef Urban. Hammering towards qed. *Journal of Formalized Reasoning*, 9(1):101–148, 2016. 7.1

[21] Sascha Bohme and Tobias Nipkow. Sledgehammer: judgement day. In *Automated Reasoning: 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010. Proceedings 5*, pages 107–121. Springer, 2010. 7.1, 7.2

[22] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022. 1.1, 3.1, 6.1, 6.1, 6.2.1

[23] 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, 2020. 1, 1.2, 1.3, 1.4, 2.1, 2, 2.4, 3.1, 4.1, 4.3, 4.5.1, 5.1, 6.1, 6.2.3, 7.2, 7.3.2

[24] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023. 1, 6.1, 6.1, 6.2.1, 6.3.2, 6.3.5

[25] Kevin Buzzard. Lean in 2024. `https://xenaproject.wordpress.com/2024/01/20/lean-in-2024/`, 2024. 7.1

[26] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting

training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. 4.1

[27] James A Carlson, Arthur Jaffe, and Andrew Wiles. *The millennium prize problems*. American Mathematical Soc., 2006. 6.1

[28] Nathan C Carter and Kenneth G Monks. Lurch: a word processor that can grade students' proofs. In *CICM Workshops*, 2013. 7.5

[29] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, 2017. 4.1

[30] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 5.1

[31] 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*, 2021. 3.5.2, 4.1, 4.5.1

[32] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022. 7.2

[33] Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V Le. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*, 2019. 1.4

[34] 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://vicuna.lmsys.org`. 1.1, 1.3, 2.1, 2.7.1, 2.7.2, 2.5b, 2.7.3, 5, 3.5.2, 5.1, 5.5.1

[35] Noam Chomsky. On the representation of form and function. 1981. 6.2.2

[36] 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. *arXiv preprint arXiv:2204.02311*, 2022. 1, 1.3, 2.1, 4.1, 4.1, 4.4, 4.5.1, 5.1

[37] Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018. 1.1

[38] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 1.1, 2.1

[39] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 3.1, 3.1

[40] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1.1, 3.5.1, 6.1

[41] Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020. 3.5.2

[42] 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. *arXiv preprint arXiv:1803.05457*, 2018. 6.2.3

[43] 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*, 2021. 6.2.2, 6.3.3

[44] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 1.4, 6.2.3

[45] Projet Coq. The coq proof assistant-reference manual. *INRIA Rocquencourt and ENS Lyon, version*, 5, 1996. 7.1

[46] Ajeya Cotra. The case for aligning narrowly superhuman models. In *AI Alignment Forum*, 2021. 6.2.1

[47] Lukasz Czajka and Cezary Kaliszyk. Hammer for coq: Automation for dependent type theory. *Journal of automated reasoning*, 61:423–453, 2018. 7.1

[48] 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. *arXiv preprint arXiv:2305.06500*, 2023. 5.3.1

[49] Databricks. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`. 1.1, 2.7.1, 3.5.1

[50] Leonardo De Moura, Soonho Kong, Jeremy Avigad, Floris Van Doorn, and Jakob von Raumer. The lean theorem prover (system description). In *Automated Deduction-CADE-25: 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings 25*, pages 378–388. Springer, 2015. 7.1, 7.3.1

[51] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023. 3.5.1

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2.1

[53] 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`. 4.4

[54] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020. 6.4.1

[55] Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*, 2022. 6.2.2

[56] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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)*, pages 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246. 6.2.2

[57] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023. 3.4, 3.5.1, 5.4, 5.5.1

[58] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR, 17–23 Jul 2022. 3.5.1

[59] Emily First. *Automating the Formal Verification of Software.* PhD thesis, 2023. URL https://scholarworks.umass.edu/dissertations_2/2812. 7.1, 7.5

[60] Emily First, Markus Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2023, page 1229–1241, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400703270. doi: 10.1145/3611643.3616243. URL https://doi.org/10.1145/3611643.3616243. 7.2

[61] Jerry A Fodor and Ernest Lepore. *The compositionality papers.* Oxford University Press, 2002. 6.2.2

[62] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based

prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2022. 6.1, 6.2.3, 6.3.2

[63] Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*, 2023. 3.1

[64] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3): 411–437, 2020. 1.1

[65] Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*, 2023. 1.2, 3.1

[66] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023. 3.1, 3.3, 6.4.2

[67] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*, 2023. 1.2, 3.1

[68] Fabian Gloeckle, Baptiste Roziere, Amaury Hayat, and Gabriel Synnaeve. Temperature-scaled large language models for lean proofstep prediction. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023. 7.1

[69] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 5.1

[70] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 5.3.1

[71] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023. 1.1, 6.4.1

[72] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts,

Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023. 3.5.1, 6.3.4

[73] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020. 4.1

[74] Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. *arXiv preprint arXiv:2102.06203*, 2021. 7.1

[75] Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=rpxJc9j04U. 7.2

[76] Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks. *arXiv preprint arXiv:2401.06751*, 2024. 6.1, 6.2.3, 6.3.2, 6.3.5

[77] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. 6.2.3

[78] Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021. 6.4.3

[79] 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*, 2021. 3.5.1, 6.2.3, 6.3, 6.3.1, 2, 6.4.2

[80] 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*, 2022. 4.1

[81] 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*, 2021. 3.5.1, 5.5.1

[82] Yinya Huang, Xiaohan Lin, Zhengying Liu, Qingxing Cao, Huajian Xin, Haiming Wang, Zhenguo Li, Linqi Song, and Xiaodan Liang. Mustard: Mastering uniform synthesis of theorem and proof data. *arXiv preprint arXiv:2402.08957*, 2024. 7.1

[83] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018. 1.1

[84] Gautier Izacard and Édouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, 2021. 4.4

[85] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022. 1.3, 4.1, 4.1, 4.5.1

[86] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. 1.3

[87] Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothee Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*, 2022. 7.1, 7.2, 7.4.1

[88] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017. 4.1, 4.1, 4.3, 4.4, 4.5.1

[89] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 3.3, 5.6

[90] Dongyeop Kang, Andrew Head, Risham Sidhu, Kyle Lo, Daniel S Weld, and Marti A Hearst. Document-level definition detection in scholarly documents: Existing models, error analyses, and future directions. *arXiv preprint arXiv:2010.05129*, 2020. 7.5

[91] 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. 5.5.2

[92] Richard M Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975. 6.1

[93] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2019. 6.2.2

[94] Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. Aligning large language models through synthetic feedback. *arXiv preprint arXiv:2305.13735*, 2023. 3.2

[95] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016. 4

[96] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022. 2.4

[97] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023. 1.1, 3.5.1, 3.5.1

[98] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. 4.1, 4.1, 4.2, 4.3, 4.4, 4.5.1

[99] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment, 2023. 2.1, 2.7.2

[100] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR, 2018. 6.2.2

[101] Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*, 2022. 4.3

[102] Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. Hypertree proof search for neural theorem proving. *Advances in neural information processing systems*, 35:26337–26349, 2022. 7.1, 7.2, 7.4.1

[103] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, et al. Obelisc: An open web-scale filtered dataset of interleaved image-text documents. *arXiv preprint arXiv:2306.16527*, 2023. 5.3.1

[104] Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in neural machine translation. 2018. 1.3

[105] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018. 1.1, 1.1, 6.2.1

[106] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 1.3, 4.1

[107] 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. *arXiv preprint arXiv:2206.14858*, 2022. 6.3.1

[108] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. 2023. 5.3.1

[109] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*, 2023. 1.2, 3.1

[110] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of

instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023. 3.5.2

[111] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 1.3, 5.1, 5.5.2

[112] Zhaoyu Li, Jialiang Sun, Logan Murphy, Qidong Su, Zenan Li, Xian Zhang, Kaiyu Yang, and Xujie Si. A survey on deep learning for theorem proving, 2024. 7.1

[113] Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and Teknium. Openorca: An open dataset of gpt augmented flan reasoning traces, 2023. 3.5.1

[114] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. 1.1, 6.1, 6.3.1, 6.3.3, 6.3.5, 6.4.2, 6.4.3

[115] Haohan Lin, Zhiqing Sun, Sean Welleck, and Yiming Yang. Lean-star: Learning to interleave thinking and proving. *arXiv preprint arXiv:2407.10040*, 2024. **??**, 8.6

[116] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021. 1.3, 2.7.2, 3.5.2

[117] Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022. 5.6

[118] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5.1

[119] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, 2017. 1.4

[120] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 1.3

[121] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning.

2023. 1.3, 5.1, 5.1, 5.5.1, 5.3, 5.5.2

[122] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. Iterative machine teaching. In *International Conference on Machine Learning*, pages 2149–2158. PMLR, 2017. 6.2.3

[123] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 5.1, 5.5.2, 5.4

[124] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023. 5.1

[125] Haley MacLeod, Cynthia L Bennett, Meredith Ringel Morris, and Edward Cutrell. Understanding blind people's experiences with computer-generated captions of social media images. In *proceedings of the 2017 CHI conference on human factors in computing systems*, pages 5988–5999, 2017. 1.3

[126] 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. *arXiv preprint arXiv:2303.17651*, 2023. 3.1

[127] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 5.3.1

[128] The mathlib Community. The lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, pages 367–381, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370974. doi: 10.1145/3372885.3373824. URL `https://doi.org/10.1145/3372885.3373824`. 7.1

[129] Mark A McDaniel, Daniel C Howard, and Gilles O Einstein. The read-recite-review study strategy: Effective and portable. *Psychological Science*, 20(4):516–522, 2009. 4.2

[130] Microsoft. Introducing the new bing, 2023. URL `https://www.bing.com/new`. 2.4

[131] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Bider-

man, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*, 2022. 1.3

[132] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023. 3.5.1, 3.5.1

[133] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 1.1

[134] Moni Naor. Evaluation may be easier than generation. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 74–83, 1996. 6.1

[135] Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*, 2020. 4.3

[136] Allen Newell and Herbert Simon. The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79, 1956. 7.1

[137] Tobias Nipkow, Markus Wenzel, and Lawrence C Paulson. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer, 2002. 7.1

[138] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021. 2.4, 7.1, 7.2

[139] National Academies of Sciences. Artificial intelligence to assist mathematical reasoning: Proceedings of a workshop, 2023. 7.1, 7.5

[140] OpenAI. OpenAI: Introducing ChatGPT, 2022. URL `https://openai.com/blog/chatgpt`. 1, 1.1, 2.1, 2.1, 2.7.1, 2.7.2, 3.1, 3.1, 3.4, 5.4, 6.1, 6.3.1

[141] OpenAI. Gpt-4 technical report, 2023. 1, 1.1, 1.1, 1.2, 1.3, 2.1, 2.1, 2.4, 2.7.1, 2.7.2, 2.7.2, 3.1, 3.1, 3.2, 3.3, 3.4, 3.4, 5.1, 6.3.1

[142] OpenAI. OpenAI: GPT-4, 2023. URL `https://openai.com/research/gpt-4`. 2.1, 6.1, 7.4.1, 7.4.3

[143] OpenAI. Model index for researchers. https://platform.openai.com/docs/model-index-for-researchers, 2023. 2.1, 2.1, 2.3, 2.7.1, 3.1

[144] 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. 1, 1.1, 2.1, 1, 2.1, 2.2, 2.7.2, 3.1, 3.2, 3.2, 3.2, 3.1, 4.5.1, 5.1, 5.3, 5.3, 5.5.1, 6.1, 6.1, 6.3.2, 6.4.1, 6.4.2

[145] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022. 3.1

[146] Vihang P Patil, Markus Hofmarcher, Marius-Constantin Dinu, Matthias Dorfer, Patrick M Blies, Johannes Brandstetter, Jose A Arjona-Medina, and Sepp Hochreiter. Align-rudder: Learning from few demonstrations by reward redistribution. *arXiv preprint arXiv:2009.14108*, 2020. 1, 2.1

[147] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, 2020. 6.2.2

[148] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070, 2021. 6.3.2

[149] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*, 2022. 3.1, 6.2.1

[150] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023. 3.3

[151] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020. 7.1, 7.2, 7.3.2, 7.3.2, 7.3.3

[152] Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. *arXiv preprint arXiv:2202.01344*, 2022. 7.2, 7.2, 7.1, 7.4.2

[153] Markus N Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. Mathematical rea-

soning via self-supervised skip-tree training. *arXiv preprint arXiv:2006.04757*, 2020. 7.1

[154] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 6.1, 6.2.3

[155] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 2.1

[156] 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*, pages 8748–8763. PMLR, 2021. 5.1, 5.5.1

[157] 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. 4.1

[158] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. 3.5.1, 6.3.4, 6.4.2

[159] 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. *arXiv preprint arXiv:2311.12022*, 2023. 1.1, 6.1, 6.2.1

[160] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, 2020. 4.1

[161] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 4.5.3

[162] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018. 1.3

[163] Subhro Roy, Shyam Upadhyay, and Dan Roth. Equation parsing: Mapping sentences to grounded equations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1097, 2016. 1.4

[164] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation

models for code. *arXiv preprint arXiv:2308.12950*, 2023. 6.3.6, 6.4.3

[165] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2021. 1.1, 3.1, 3.5.1, 6.1, 6.4.1

[166] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022. 1.1, 6.1, 6.2.1

[167] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 1.3

[168] John Schulman. Reinforcement learning from human feedback: Progress and challenges. In *Berkley Electrical Engineering and Computer Sciences. URL: https://eecs. berkeley. edu/research/colloquium/230419 [accessed 2023-11-15]*, 2023. 5.6

[169] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3.5.1, 5.5.1, 6.3.4

[170] Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34:6695–6706, 2021. 6.1

[171] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022. 5.1

[172] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023. 6.1, 6.2.1

[173] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023. 1.3

[174] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 6.3.3

[175] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023. 6.3.4, 7.1, 7.2, 7.3.2, 7.4.2

[176] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35: 9460–9471, 2022. 3.1

[177] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. 2.7.2

[178] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. 1, 1.1, 3.2, 3.4, 5.1, 5.3, 5.4, 6.1, 6.3.2, 6.4.1

[179] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*, 2022. **??**

[180] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023. **??**, 8.4

[181] Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Salmon: Self-alignment with principle-following reward models. *arXiv preprint arXiv:2310.05910*, 2023. **??**, 1.2, 8.2

[182] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*, 2023. **??**, 1.2, 3.1, 3.1, 3.2, 3.5.1, 3.6, 5.4, 6.4.1, 8.1

[183] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-cqvvvb-NkI. 5, 2.4, 8.3

[184] Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. *arXiv preprint arXiv:2403.09472*, 2024. **??**, 8.5

[185] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022. 3.5.2

[186] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, 2020. 6.1, 6.2.3, 6.3.2

[187] Christian Szegedy. A promising path towards autoformalization and general artificial intelligence. In *Intelligent Computer Mathematics: 13th International Conference, CICM 2020, Bertinoro, Italy, July 26–31, 2020, Proceedings 13*, pages 3–20. Springer, 2020. 7.1, 7.5

[188] Terence Tao. https://mathstodon.xyz/@tao/111287749336059662, 2023. 1

[189] 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. 1.1, 1.3, 1, 2.1, 2.3, 2.7.1

[190] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022. 4.1, 4.5.1

[191] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[192] Amitayush Thakur, Yeming Wen, and Swarat Chaudhuri. A language-agent approach to formal theorem-proving. *arXiv preprint arXiv:2310.04353*, 2023. 7.2, 7.1, 7.4.2

[193] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1.3, 2.1, 2.7.1, 2.4, 3.1, 5.5.1

[194] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1.1, 1.2, 1.3, 3.1, 3.1, 3.2, 3.3, 3.5.1, 3.5.1, 3.5.1, 3.5.2, 3.2, 3.5.3, 5.1, 5.3, 5.3.1, 3, 5.4, 6.3.6

[195] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024. 6.1

[196] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. 6.1, 6.3.3, 6.3.4, 6.4.1, 6.4.2

[197] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning, 2023. 1.1, 2.1

[198] Haiming Wang, Huajian Xin, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, Jian Yin, Zhenguo Li, Heng Liao, and Xiaodan Liang. Lego-prover: Neural theorem proving with growing libraries, 2023. 7.2

[199] Haiming Wang, Ye Yuan, Zhengying Liu, Jianhao Shen, Yichun Yin, Jing Xiong, Enze Xie, Han Shi, Yujun Li, Lin Li, et al. Dt-solver: Automated theorem proving with dynamic-tree sampling guided by proof-level value function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12632–12646, 2023. 7.2

[200] Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR, abs/2312.08935*, 2023. 6.1, 6.3.1, 6.3.3, 6.2

[201] Qingxiang Wang, Chad Brown, Cezary Kaliszyk, and Josef Urban. Exploration of neural machine translation in autoformalization of mathematics in mizar. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 85–98, 2020. 7.1

[202] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*, 2022. 4.5.1

[203] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 1.1, 1.2, 1, 2.1, 1, 2.3, 2.4, 3.1, 4.3, 6.3.2, 6.3.4, 6.4.1

[204] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022. 1.1

[205] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023. 3.5.2, 3.2, 5.5.2

[206] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021. 1.1, 3.5.1, 6.1, 6.4.1

[207] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022. 1.4, 5, 2.4, 4.3, 6.3.1, 7.1, 7.2

[208] Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*, 2023. 6.2.1

[209] Sean Welleck and Rahul Saha. Llmstep: Llm proofstep suggestions in lean. *arXiv preprint arXiv:2310.18457*, 2023. 7.2, 7.3.3, 7.4.1

[210] 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. 7.1

[211] Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. *Advances in Neural Information Processing Systems*, 35:4913–4927, 2022. 7.1

[212] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021. 6.1

[213] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024. **??**

[214] Yuhuai Wu, Markus N Rabe, Wenda Li, Jimmy Ba, Roger B Grosse, and Christian Szegedy. Lime: Learning inductive bias for primitives of mathematical reasoning. In *International Conference on Machine Learning*, pages 11251–11262. PMLR, 2021. 7.1

[215] Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. *Advances in Neural Information Processing Systems*, 35:32353–32368, 2022. 7.1, 7.2

[216] Huajian Xin, Haiming Wang, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, et al. Lego-prover: Neural theorem proving with growing libraries. *arXiv preprint arXiv:2310.00656*, 2023. 7.4.1

[217] 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. 3.5.1

[218] 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. 7.1, 7.2, 7.3.3, 7.4, 7.1, 7.4.1

[219] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023. 3.2

[220] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, 2019. 1

[221] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing, pages 2369–2380, 2018. 4.1, 4.1, 4.5.1

[222] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. 5

[223] Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. Internlm-math: Open math large language models toward verifiable reasoning, 2024. 7.1, 7.4, 7.1, 7.2, 7.4.1, 7.4.2, 7.4.3

[224] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 5.3.1

[225] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 5.1

[226] Fei Yu, Anningzhe Gao, and Benyou Wang. Outcome-supervised verifiers for planning in mathematical reasoning. *arXiv preprint arXiv:2311.09724*, 2023. 6.1, 6.3.3

[227] 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. 6.1, 6.3.1

[228] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. 7.1, 7.2, 7.4.2

[229] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 4.1, 4.5.1

[230] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023. 1.3

[231] Xueliang Zhao, Wenda Li, and Lingpeng Kong. Decomposing the enigma: Subgoal-based

demonstration learning for formal theorem proving, 2024. URL https://openreview.net/forum?id=xLoxMvO695. 7.2

[232] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023. 3.5.1

[233] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021. 4.5.3, 6.4.1

[234] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021. 7.1, 7.4, 7.4.1

[235] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. 3.1, 3.4, 3.5.2, 5.4

[236] Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. Goal driven discovery of distributional differences via language descriptions. *arXiv preprint arXiv:2302.14233*, 2023. 1

[237] Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*, 2020. 1.3

[238] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023. 1.1, 3.1, 3.1, 5.3.1

[239] Denny Zhou. Teach language models to reason. https://dennyzhou.github.io/LLMs-Reason-Taiwan-2023.pdf, 2023. Accessed: 2024-05-21. 7.1

[240] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2022. 1.1, 6.1, 6.2.2, 6.3.2

[241] Jin Peng Zhou, Charles Staats, Wenda Li, Christian Szegedy, Kilian Q Weinberger, and

Yuhuai Wu. Don't trust: Verify–grounding llm quantitative reasoning with autoformalization. *arXiv preprint arXiv:2403.18120*, 2024. 7.1

[242] Jin Peng Zhou, Charles E Staats, Wenda Li, Christian Szegedy, Kilian Q Weinberger, and Yuhuai Wu. Don't trust: Verify – grounding LLM quantitative reasoning with autoformalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=V5tdi14ple. 7.2

[243] 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*, 2023. 5.1

[244] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. 3.1, 3.2, 5.1, 5.3, 6.1