Towards Grounded Multimodal Enterprise Document Understanding

Armineh Nourbakhsh

CMU-LTI-25-007

April 2025

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

Thesis Committee:

Carolyn P. Rosé, Chair
Sameena Shah (J.P. Morgan Chase & Co.), Co-Chair
Eric H. Nyberg
Matthew R. Gormley
Mohit Bansal (University of North Carolina at Chapel Hill)

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

Keywords: visually rich document understanding, visual question answering, form understanding, multimodal reasoning, quantitative reasoning, question answering over tabular data, grounded question answering, multimodal groundedness

To Arman

Abstract

Document-grounded workflows drive operational efficiency in many enterprise domains. In Finance, onboarding and offboarding of clients, monitoring of client activities, assignment of risk, assessment of credit, and other integral functions are dependent on processing documents across a wide variety of categories and formats, including business filings, financial reports, tax forms, legal contracts, invoices, payment records, and other disclosures. The document understanding tasks associated with these processes encompass several multimodal reasoning challenges, including spatial, visual, and quantitative reasoning.

Against this backdrop, investment in AI-augmented workflows has grown rapidly over the past decade [59, 60]. In highly regulated industries such as Finance, such workflows are expected to comply with requirements related to performance and robustness, including the maintenance of comprehensive data lineage. This means that an Information Extraction model is required to provide datapoints that are fully traceable back to the context from which they were extracted. Groundedness has major implications for downstream applications, as it can improve explainability, expose the provenance of the output, and enhance human-AI interaction.

In recent years, multimodal (large) language models have emerged as a promising approach to document understanding. While these models have demonstrated better overall performance across several tasks, their decoder-based, generative architecture leaves them open to poor groundedness (if not hallucinations), and makes it difficult to localize their outputs. This has led to challenges related to groundedness (or lack thereof) in tasks such as Key Information Extraction and extractive Visual Question Answering, which has in turn complicated the adoption of such models in production pipelines, especially when the requirements for reliability and explainability outweigh performance.

This work addresses the challenge of groundedness in multimodal enterprise document understanding in the context of two prominent reasoning tasks, namely, quantitative reasoning and spatio-visual reasoning. We demonstrate how we can enhance the performance, robustness, and generalizability of models by improving their grounding within the input. In quantitative reasoning, we show how grounding the model in numerical language can enhance compositional generalization, a key challenge in robustness and OOD performance. We further demonstrate how spatio-visual reasoning can be grounded in the layout and structure of a document, leading to more efficient and robust multimodal models.

Concretely, we introduce three new methods to the field of grounded multimodal enterprise document understanding: 1) A new mechanism to attend to fine-grained components of the input that express arithmetic operations, hence improving compositional generalization in quantitative reasoning tasks. 2) A metric-learning strategy that is grounded in counterfactually-associated samples, and leads to more robust and generalizable quantitative reasoning models. 3) A topological representation of documents that enhances performance on several multimodal tasks by grounding textual reasoning within the spatial layout of each page. We tie these methods together by proposing an evaluation strategy that accounts for fine-grained spatial and contextual grounding in a visual question-answering task. Using Visual Question Answering as an umbrella task, we demonstrate how our evaluation framework can expose shortcomings in spatio-visual and quantitative reasoning, especially when compared against human performance.

Acknowledgments

This work would not have been possible without the mentorship, support, and contributions of numerous researchers at the Language Technologies Institute, JP-Morgan Chase & Co., and beyond.

Firstly, I would like to thank my advisor, Carolyn Rosé, who has been a role model in constructive mentorship, and whose support and guidance were indispensable to the completion of my studies. I consider myself very fortunate to have had the opportunity to study under Carolyn's supervision.

I am very grateful to my committee members, Sameena Shah, Eric Nyberg, Matthew Gormley, and Mohit Bansal for being generous with their time and knowledge despite the hectic timeline of my studies. Every committee member has had a major impact on the course of my studies—Sameena Shah for encouraging me to ground my work in real-world enterprise applications, Eric Nyberg for his guidance on the challenges related to Question Answering, Matthew Gormley for his feedback on the design of generative architectures, and Mohit Bansal for encouraging me to explore generative approaches to multimodal reasoning.

I have also had the privilege of collaborating with very talented students and colleagues, whose contributions made the studies presented in this manuscript possible. Thank you, Zhao (Judy) Jin, Siddharth Parekh, Cathy Jiao, Pranav Shetty, Sireesh Gururaja, Rosanna Vitiello, Ivan Brugere, and all other researchers and colleagues who provided feedback and input throughout my studies.

Additionally, I have benefited tremendously from the mentorship of Jill Fain Lehman (LTI), Chas Murray (LTI), Natraj Raman (JP Morgan & Co.), and Hao-Ren (Howard) Yao (NIH) all of whom have the gift of infinite curiosity and passion that I hope to emulate.

Lastly, I would like to thank Manuela Veloso and JPMorgan Chase & Co. for giving me the opportunity to pursue a degree at the Language Technologies Institute. As a mid-career researcher who learned the basics of research by following the work of LTI faculty, this was nothing short of a dream come true, and a privilege that I will forever cherish.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multimodal document understanding, also known as Visually Rich Document Understanding (VRDU), and the tasks that it encompasses—including Key Information Extraction (KIE), Relation Extraction (RE), Visual Question Answering (VQA), and Quantitative Question Answering (QQA), to name a few—constitute a major operational bottleneck in enterprise settings [115, 129]. Due to their rich structure, length, domain-specific language, hybrid numeric/textual content, and spatio-visual complexity, enterprise documents such as reports, memos, invoices, forms, and contracts are often processed using human supervision. This manual process is cumbersome and therefore error prone, so much so that some institutions are compelled to adopt a dual review protocol for the task of information extraction and data entry [18]. This, coupled with the large volume of documents in many enterprise settings[1] has led to a substantial and growing demand for Document Intelligence services [59, 60].

Against this backdrop, researchers in the domain of VRDU have developed a suite of benchmarks that examine the performance of SotA models on KIE [57, 63, 128, 166], RE [63, 128], VQA [113, 114, 179], and QQA [26, 27, 224, 227]. The advent of Multimodal Generative (Large) Language Models has moved SotA performance across these tasks to new frontiers, in some cases coming close to human performance. As of February 2025, the top 5 performers on the DocVQA[2], InfographicsVQA[3], and TAT-QA[4] leaderboards are Generative LLMs. The top performing model on DocVQA [11] is within 2 points of human performance, and the top-performing model on TAT-QA [228] within 3 points.

Despite positive contributions to the VRDU domain, the popularity of generative models has moved the field away from producing grounded outputs, as we will argue in Chapter 2. This means that it is not always possible to determine the lineage of each output token with respect to the input. For extractive tasks such as KIE and Extractive QA, this means that the output cannot be deterministically traced back to where it was extracted from. This has constrained the adoption of multimodal document understanding models in enterprise settings, where regulatory

[1] As an example, in 2024, J.P. Morgan Chase served nearly 80 million consumers [19]. Considering each customer's records, filings, tax forms, identification documents, and other disclosures, the volume of documentation in the retail banking business alone could scale to hundreds of millions.

[2] https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1

[3] https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=3

[4] https://nextplusplus.github.io/TAT-QA/

guidelines often require data lineage to be established in great detail [83].

The goal of our research is to propose ***grounded methodologies*** for the task of multimodal enterprise document understanding. This not only enables the operationalization of VRDU models in enterprise settings, but, as we will demonstrate, enhances the robustness, efficiency, and generalizeability of such models. Toward this goal, we tackle grounded reasoning in two relevant subdomains, namely, *quantitative reasoning*, and *spatio-visual reasoning*. Before we explore these domains in detail, we examine the role of groundedness in enterprise settings, not only to identify key challenges, but also to establish the motivation behind our work.

1.1 Groundedness in enterprise settings

Why should groundedness be uniquely important to enterprise applications? What sets enterprise Document AI apart from other applications of the technology in other domains? In this section, we address this question using a scenario that is inspired by a common real-world task.

Suppose Alice is a knowledge worker at a financial institution, and is in charge of reviewing client documents. Each institutional client provides a document with a list of authorized signatories, their titles, contact information, and signature samples. This list is later used to verify whether legally binding agreements have been signed by authorized stakeholders. As part of a remediation project, Alice is tasked with reviewing 1,000 authorized signatory forms, extracting key information, and keying them into a new database.

To make the task more manageable, Alice would like to reduce her manual workload by 70%, either by reviewing only 30% of the documents, or by reviewing only 30% of each document's contents, and delegating the remainder of the work to an automated solution. This would require the automated solution to perform the following tasks:

1. Process each authorized signatory document and extract the following information: names, metadata (such as titles, addresses, and contact information), and signature samples. This is associated with the task of *Key Entity Extraction (KIE)*.
2. Associate all attributes related to the same entity. For example, the name, contact information, and signature sample of each individual should be grouped together. This is associated with the task of *Relation Extraction (RE)*
3. Map the information about each entity into a schema that the new database recognizes.
4. Create a detailed trace of where each piece of information was extracted from so that auditors can verify that proper protocol was followed. This is associated with the task of *Localization*.

The top row of Table 1.1 illustrates the output that Alice expects from an automated system. Ideally, each entity must be tagged within the document so that the extracted entity can be mapped to a bounding box within the page. In VRDU literature, this is known as *Localization*. The model also needs to be able to group each signatory's name and corresponding title and signature. This is often addressed through the task of *Relation Extraction (RE)*. Lastly, each extraction (or ideally, each grouping, each page, or each document) needs to have a confidence score that Alice can use to decide whether she needs to review the model's output for accuracy.

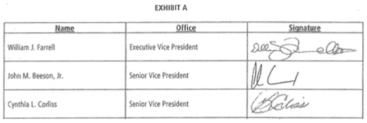
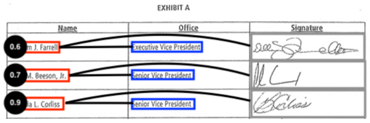
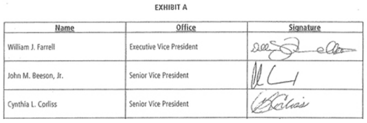
 EXHIBIT A Name | Office | Signature William J. Farrell | Executive Vice President John M. Beeson, Jr. | Senior Vice President Cynthia L. Corliss | Senior Vice President	 EXHIBIT A Name Office Signature 0.6 m J. Farrell Executive Vice President 0.7 M. Beeson, Jr. Senior Vice President 0.9 a L. Corliss Senior Vice President	**Ideal output:** Each signatory, title, and signature box is extracted and tagged within the document [red, blue, and gray boxes]. Each signatory is used as an anchor to group the metadata [black links]. Each entity (or grouping) is assigned a confidence score [black circles].
 EXHIBIT A Name | Office | Signature William J. Farrell | Executive Vice President John M. Beeson, Jr. | Senior Vice President Cynthia L. Corliss | Senior Vice President	William J. Farrel: NAME0 Executive Vice President: TITLE0 bbox{30, 10, 50, 25}: SIG0 [NAME0, TITLE0]: LINK0 John M. Beeson, Jr.: NAME1 Senior Vice President: TITLE1 bbox{30, 30, 50, 45}: SIG1 [NAME1, TITLE1]: LINK1 ...	**Common output:** Extractions may not be associated with bounding boxes. Different solutions may be needed for text vs. handwriting. Links may be unavailable or only partially available. Confidence scores are often not available (or log-probs are uncalibrated).

Table 1.1: Top row: The expected output of a VRDU model when processing an authorized signatory document. Bottom row: The output often generated by SotA approaches. Note that due to the confidentiality of authorized signatory forms, we have used a public example from a credit agreement [28].

This is related to the problem of *model calibration*.[5]

Let us go through the above-mentioned tasks and examine their relationship to the concept of groundedness.

Relation Extraction (RE) is the associative counterpart to the task of Key Entity Extraction (KIE), one of the most popular tasks in the VRDU literature. As our scenario demonstrates, in many real-world settings, KIE and RE need to be performed in tandem to enable end-to-end automation. In our scenario, a model that is solely trained on KIE would be able to identify each authorized signatory, phone number, and address, but would not be able to group them together or map them to the relational schema of a database.

Despite their relevance to real-world applications, associative tasks are often ignored in VRDU datasets, possibly due to the high cost of annotating documents for multiple tasks. Datasets such as FUNSD [63], CORD [128], DocILE [162], and BuDDIE [233] include some relation annotations, but only one (FUNSD) covers complete hierarchical relations in addition to key-value pairings. FUNSD also happens to be the smallest dataset, covering only 199 samples. This has led to an under-representation of RE in research publications, as we will see in Section 4.2. As part of our research, we will demonstrate that spatially-grounded models, i.e. those that are designed to effectively capture complex layouts, are able to capture relations between different entities more effectively.

Groundedness (or lack thereof) poses additional challenges to the applicability of VRDU models in enterprise settings. Let us once again consider our scenario. As stated earlier, Alice would like to reduce her workload by 70%. Let us suppose that she is able to find a SotA model

[5] Note that the above requirements are not limited to our particular scenario and generalize to most information extraction tasks in enterprise settings. Some requirements (such as traceability of output) extend to tasks beyond information extraction, such as question answering over documents, and summarization.

that has an F1 score of 0.99 across all KIE benchmarks. For simplicity, we will assume that this means the model makes one mistake per 100 extractions. If Alice applies the model to the signatory forms, there will likely be errors, given that 1,000 forms are likely to include more than 100 signatories. If Alice is not able to locate the possible errors, she will have to review every one of the model's extractions to verify its accuracy. Assuming that Alice can perform the verification task faster than the extraction task, we will estimate her time-saving as 50%.[6] This will still not meet her target of 70% of documents (or 70% of fields) being processed in a "straight-through" fashion without a manual touchpoint. In order for Alice to reach her target, she would need a model that is *well calibrated*, and can indicate which documents or which contexts are likely to include errors.

Despite the recent attention that calibration research has attracted with regard to the detection of hallucinations in LLM outputs, the VRDU literature has remained largely focused on performance without much regard for calibration. As we will see in Section 4.2.4, models that are carefully designed to ground their reasoning in the multimodal signal can not only achieve better performance, but also produce lower calibration error. Overall, our research will demonstrate that enterprise document understanding models can be designed, trained, and evaluated with groundedness in mind, resulting in models that are more parameter and data efficient, generalize to out-of-distribution samples, and generate better calibrated outputs.

1.2 Enhancing groundedness in enterprise document understanding

Our investigation into improving groundedness for multimodal enterprise document understanding is organized into three subdomains, namely, quantitative reasoning, spati-visual reasoning, and model evaluation. The following sub-sections introduce each domain and the relevance of groundedness within the VRDU literature in that domain.

1.2.1 Quantitative reasoning

Quantitative reasoning encompasses a wide range of research areas including numeracy [168, 172, 173], quantitative grounding of language models [149, 163, 164], solving math word problems [6, 193, 218], and question answering over tabular data [26, 27, 224, 227]. Each field has attempted to take advantage of mathematical, arithmetic, and algebraic knowledge that governs the reasoning required to perform quantitative tasks. Some studies have attempted to create models that exhibit knowledge about magnitudes and are able to compare various quantities [172, 221]. Others have pursued more explicitly symbolic approaches [148, 149].

In multimodal document understanding, quantitative reasoning is focused on hybrid tabular/-text contexts, and lies at the intersection of spatial and numerical reasoning. Several recently

[6] If Alice is following a dual review process (i.e. a Maker-Checker process), then the 50% time-saving estimate is consistent with removing the Maker from the process, allowing Alice to act as the Checker. Having said that, the estimate is still likely to be very generous, because ungrounded models do not contextualize their extractions, and Alice would need to manually locate each extracted output in the original document before verifying it.

published datasets aim to tackle this particular problem [26, 27, 93, 224, 227]. Table 1.2 illustrates a simple example of a question answering task over tabular data.

<table>
<tr><td><b>Question</b></td><td>What was the net change in revenue from 2019 to 2020?</td></tr>
<tr><td><b>Tabular context</b></td><td><table>
<tr><th>Metric ($M)</th><th>2018</th><th>2019</th><th>2020</th></tr>
<tr><td>Operating expenses</td><td>35</td><td>29</td><td>30</td></tr>
<tr><td>Revenue</td><td>70</td><td>80</td><td>60</td></tr>
</table></td></tr>
<tr><td><b>Verbalized facts</b></td><td>2019 revenue was $80M.
2020 revenue was $60M.</td></tr>
<tr><td><b>Output program</b></td><td><code>subtract(80, 60)</code></td></tr>
<tr><td><b>Answer</b></td><td><code>-20</code></td></tr>
</table>

Table 1.2: Example of a quantitative QA problem over tabular data.

One key challenge that is fundamental to developing robust models in this domain is the challenge of compositional generalization [119, 125, 212]. Beginning with simple concepts and primitives, humans are able to compose more complex concepts and use them to tackle sophisticated reasoning tasks that require multi-step calculations. This has inspired some researchers to emulate some of the strategies that humans use in order to learn these compositions. As an example, chain of thought prompting has successfully been used to encourage models to break down complex problems into smaller steps before solving them [197].

Another approach is to address gaps in the model's learning by generating "what if" scenarios. Consider the sentence:

"5 plus 3 equals 8.".

If this is the only example of addition that the model encounters during training, it might memorize operands such as 5 and 3 as signifying an addition. This is fundamentally a challenge of grounding, as the model does not learn to ground its reasoning in the correct expressions. By generating examples that perturb operands or operators, the model can be encouraged to capture semantics at the component level [94]:

What if instead of "5" we used "2"? → "2 plus 3 equals 8."
What if instead of "plus" we used "minus"? → "5 minus 3 equals 2."

Research on compositional generalization has shown promise, but SotA models still lag behind human performance [94], mainly limited by the inflexibility of the data augmentation methods mentioned above. In our research, we develop more robust models by exploiting the correspondence between natural language terms and quantitative semantics in a more explicit fashion, i.e. by grounding the model's reasoning in natural expressions of arithmetic operations.

1.2.2 Spatio-visual reasoning

As we will lay out in Section 2.2.2, two transformer-based neural architectures [182] dominate SotA VRDU benchmarks: *Encoder-based models* approach visual reasoning the same way it is tackled in open-domain Visual Question Answering (VQA). A neural language model is used to generate contextual embeddings for the text in a document. Independently, a visual feature extractor (usually a Convolutional Neural Network [46, 151] or a vision transformer [37]) is used to generate visual embeddings. The two are then fused and trained in an end to end fashion [9, 95, 205]. *Decoder-based models* follow the same principle, but use the generative objectives of Large Language Models [11, 23, 104, 190].

Decoder-based models are prone to hallucinations [55, 65] and do not guarantee grounding in their outputs [232]. Encoder-based models, while easier to ground, often produce poorly-calibrated output probabilities, as we will show in Section 4.2.4.

The grounding challenge can only be addressed by developing models that exploit how visual information is displayed in a document. These signals are often not present in open-domain images, and therefore not exploited by popular image encoders. Consequently, VRDU models that use these image encoders as their visual backbone fail to ground their spatio-visual reasoning in the layout and structure of the input document.

Documents often follow a grid system of layout where horizontal and vertical alignments play an important role in organizing information on a page. Visual contrast also helps the reader navigate the information easily. A small study by Nguyen et al. [122] showed that when presented with plain text, humans were 60% slower in finding relevant information for a question answering task than when presented with the full layout.

The literature on layout design presents four key principles that govern how readers navigate the information on a page [75]:

- **Contrast**: Differences in font face, size, color, or background can signal correspondence between different elements, or a hierarchical relationship. Figure 1.1 illustrates how contrast can indicate a title/content relationship.
- **Proximity**: Elements that lay close to each other often have some semantic correspondence. Figure 1.1 illustrates how proximate segments can form a block. Studies such as Raman et al. [145] have demonstrated how visual attention maps utilize the spacing or gaps between elements to determine structures such as blocks and columns.
- **Alignment**: Vertical or horizontal alignment is used extensively in constructs such as tables, lists, or infographics, as shown in Figure 1.1.
- **Repetition**: Consistency is a key component of layout design. If a certain font or size is used for one footnote, it is likely that other footnotes will follow the same style. The size difference between headings and sub-headings is the same throughout the document, and so on. This principle ensures that each document follows a fixed "template" where a limited set of rules, shapes, and colors govern the layout. This can be of great advantage to automation efforts, since it limits the scope of features and their interactions.

There have been some recent efforts in accounting for these layout design principles in VRDU models. Graph representations, covered in Section 2.2.1, attempt to capture the grid-like layout of each page. Visual feature extraction networks, covered in Section 2.2.2, attempt to use the visual signal to split each page into segments. Nevertheless, a more deliberate approach is needed to fully exploit the advantages these principles offer. Throughout our studies, we will demonstrate how models that are designed explicitly to draw on these principles produce better-grounded outputs.

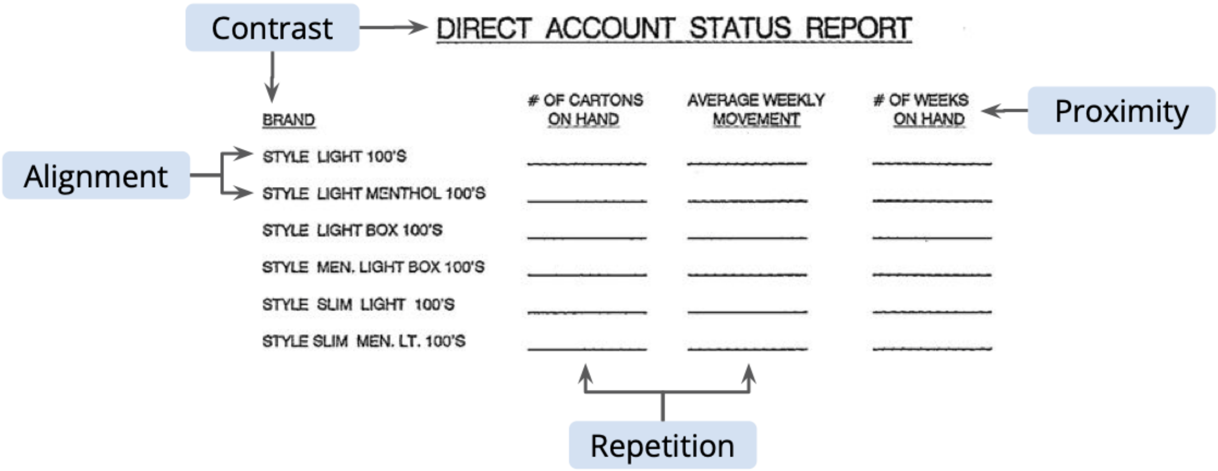


Figure 1.1: Four key principles of layout design, illustrated in a form extracted from the FUNSD dataset [64].

1.2.3 Model evaluation

Similar to most other research fields, the field of VRDU has been driven by popular benchmarks and the standard evaluation metrics that they employ. In generative tasks such as question answering, most such evaluation metrics rely on the surface similarity between ground truth and predicted answers. This does not bode well for groundedness, as surface similarity can be a poor indicator for correctness or robustness.

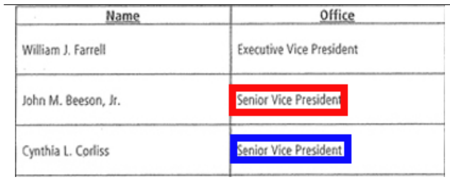
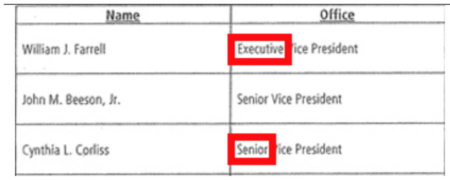
Document	Question: `What is the title of Cynthia L. Corliss?`	
 Name | Office William J. Farrell | Executive Vice President John M. Beeson, Jr. | Senior Vice President Cynthia L. Corliss | Senior Vice President	**Model A** Answer: `Senior Vice President` ANLS: 1.0	**Model B** Answer: `Executive Vice President` ANLS: 0.625
Document	Question: `Is Cynthia L. Corliss a senior executive?`	
 Name | Office William J. Farrell | Executive Vice President John M. Beeson, Jr. | Senior Vice President Cynthia L. Corliss | Senior Vice President	**Model A** Answer: `No` ANLS: 1.0	**Model B** Answer: `Yes` ANLS: 0.0

Table 1.3: An example illustrating how lack of grounding can lead to misleading assessments of a model's performance. Top-row: Extractive QA. Bottom row: Abstractive QA. The image is excerpted from [28].

Let us illustrate this using an example from the top row of Table 1.3. Given an authorized signatory form and the question "What is the title of Cynthia L. Corliss?" two hypothetical models are shown to provide ungrounded answers. The models are evaluated using Average Normalized Levenshtein Similarity, popularized by the DocVQA benchmark [113]. Model A produces the correct output with a perfect score, but without any grounding information, it is unclear whether the output refers to the proper bounding box (blue) or is based on the incorrect context (red). Model B produces an incorrect response, referring to the title of another signatory. Nevertheless, the ANLS metric is calculated at 0.625 due to a partial match with the gold answer.

In 8.5% of the training samples in DocVQA, there are two or more instances of the gold answer within the input page, making it difficult to properly contextualize the answers in a post-processing step. Partial matches only complicate this problem further.

Another challenge arises from the lack of grounding for abstractive questions, despite the requirement in many enterprise settings that every abstractive decision needs to be explicitly evidenced. Consider the second row of Table 1.3 which shows an example of an abstractive Yes/No question. To determine whether "Cynthia L. Carliss" is a senior executive, a model would need to follow a particular reasoning path, first locating her title on the page, and then mapping it to a collection of possible roles that qualify as senior executive titles.[7] In the absence of any grounding or explanation, it would be unclear whether a model is providing the correct answer ("No") or the incorrect answer ("Yes") based on a simple match with the keywords "Senior" and "Executive", respectively. While grounded reasoning datasets exist in the unimodal literature [26, 220] and in open-domain VQA [135, 229], such datasets are yet to be popularized in multimodal document understanding.

In Chapter 5 we propose an evaluation framework that accounts for localization and groundedness of predicted answer. We demonstrate that our score is better correlated with the calibratedness and robustness of models, enabling downstream practitioners to better assess the reliability of VRDU models within their domain.

1.3 Overview of the dissertation

The ultimate goal of our research is three-fold:

1. To address current gaps in research towards grounded document layout understanding and quantitative reasoning.
2. To propose grounding methodologies that improve the performance, efficiency, and robustness of current models.
3. To propose evaluation metrics that account for groundedness as a key metric.

In the next chapter, in addition to reviewing the current literature on document understanding, we offer a high-level categorization of how groundedness can be encouraged in VRDU models. These three categories are:

- *Designing* neural architectures that are grounded in relevant semantic signals.
- Creating innovative (unsupervised or self-supervised) *objectives* that ground the model in the input.
- Changing the problem *search space* by curating training samples, or scaffolding the space in an explicit fashion.

[7] In many enterprise settings, such knowledge bases and taxonomies are available as part of training material, policy documents, business rulesets, or structured databases.

Table 1.4 maps out this framework in relation to the multimodal signals that are relevant to enterprise document understanding. This framework will inform how we present our research contributions.

Reasoning	**Grounding the design**	**Grounding the objective**	**Grounding the search space**
Quantitative	Design models that recognize numerical relations	Learn quantitative compositions	Augment the search space with quantitative knowledge
Spatio-visual	Design models that follow the document layout	Learn to construct realistic document layouts	Scaffold the search space according to layout signals

Table 1.4: An overview of two multimodal reasoning areas and a summary of how grounding is often encouraged in each area.

The remainder of this document will map out the background and our research contributions. Below is a summary of what the remaining chapters of this document cover:

- Chapter 2 will review the current literature. The chapter will conclude by a summary of SotA art models and how groundedness is incorporated (or left out) in their design and training paradigm.
- Chapter 3 will cover our contributions to quantitative reasoning, specifically to compositional generalization for multi-step reasoning tasks:
    - In Section 3.1, we introduce an attention mechanism that is grounded in the quantitative semantics of natural language, and demonstrate that this leads to better compositional generalization.
    - In Section 3.2, we extend our work to create metric-learning objectives that encourage better-grounded representations, leading to models that generalize to OOD samples.
- In Chapter 4, we introduce grounded methodologies for spatio-visual reasoning of complex enterprise documents for the tasks of Key Information Extraction (KIE) and Relation Extraction (RE).
    - Section 4.1 demonstrates how graph-based structures can facilitate the representation of complex layouts, and encourage VRDU models to capture the design principles introduced in Section 1.2.2.
    - Section 4.2 extends this idea by proposing a generative graph neural network, showing improvements in both grounding and calibration against SotA VRDU models.
- Chapter 5 proposes a new evaluation framework for Document VQA models that measures the groundedness of their outputs, enabling enterprise users to investigate the utility of each model for grounded applications.
- Chapter 6 provides an overview of our work and lays out future directions for research into grounded enterprise VRDU.

Chapter 2

Background

As mentioned in the previous chapter, enterprise VRDU models need to accommodate two categories of signal that go beyond textual semantics:

- ***Quantitative signal***, which indicates content that requires discrete or numerical reasoning.
- ***Spatio-visual signal***, which encompasses layout indicators such as distances and alignments among textual segments, as well as style, color, and other visual indicators.

Holistic understanding of an enterprise document requires modeling these aspects cohesively, and extensive research has been devoted to addressing this challenge. Inspired by multimodal neural language models, many studies have treated these signals as multimodal features that are encoded individually and subsequently fused with textual representations. In contrast, certain studies have incorporated these signals as *contextual grounding* for textual semantics. As an example, instead of encoding spatial, visual, and textual features separately and fusing them afterward, some studies have incorporated the spatial signal as additive or multiplicative augmentation for text representations. This form of contextual grounding has shown increasing promise in the field of VRDU.

In this chapter, we will provide an overview of studies that have explored grounded multimodal reasoning over documents. As previously mentioned, most such studies have encouraged multimodal grounding by extending the *design*, modifying the *training objectives*, or scaffolding the *search space* of modern language models such that they can accommodate multimodal signals more effectively. We therefore present our review of the literature in the context of the above methods. Table 2.1 shows an overview of the two types of multimodal signals against the three common approaches to grounding VRDU models in those signals. Each cell in the table summarizes one or more common methodologies that are employed by today's SotA models. The following sections will describe these methodologies in more detail.

2.1 Grounded quantitative reasoning

Quantitative reasoning is a key but particularly challenging aspect of holistic document understanding. In contrast to spatial and visual signal both of which can be modeled separately from the text and fused afterwards, quantitative signal can be expressed in a non-symbolic fashion that makes it difficult to separate from the semantics of the text. Expressions that convey quantities,

Reasoning	Grounding the design	Grounding the objective	Grounding the search space
Quantitative	Reason over graphs	Learn aligned patterns	Augment input with long-tail samples
Spatio-visual	- Augment or trim attention patterns according to document layout - Use modality-aware fusion	- Learn to recover masked components in the layout - Learn masked vision-language modeling	- Generate according to intra-component relationships - Augment visual input to cover common gaps in the search space

Table 2.1: A summary of two meta-textual reasoning areas and common approaches that are used to encourage groundedness in them.

metrics, or arithmetic operations, need to be processed seamlessly together with text.

Some studies have explored language models that exhibit numeric literacy by modeling magnitude and polarity [154, 172, 173]. But numeracy alone does not suffice for quantitative reasoning tasks in all contexts. Consider the below sentence:

> "After a deductible of $600, for a co-pay of ten percent, your out of pocket cost is $200."

Even though "$600", "ten percent", and "$200" are all expressions that allow accurate estimations of magnitude and scale, they need to be contextualized properly with the concepts of "deductible", "co-pay", "out of pocket cost", all of which carry quantitative semantics. Language models that have been trained for numeracy can capture the magnitude of each value and perform comparative analyses to answer questions such as "Is the deductible higher than the out of pocket cost?" They may also be able to do magnitude estimation to answer questions such as "Does $600 sounds like a reasonable deductible?" But they may fail to perform complex quantitative reasoning required to answer questions such as "What was the original cost of the procedure?"

In tabular question answering tasks where numeric values are segregated in tabular structures, maintaining the semantic link between numbers and the textual context surrounding them introduces an added challenge. In fact, a hybrid table/text context may be even more difficult to tackle than an exclusively tabular or textual context [26].

Studies that have tackled quantitative reasoning in QA tasks fall into two categories. Some studies have explored quantitative reasoning for answering questions over real-world data such as statistical records [27], Wiki entries [22], enterprise documents [70], and financial reports [227]. Since numeric data is very often expressed in tabular structures, this category often involves question answering over tabular data, or hybrid table/text passages. Other studies have explored Math Word Problems [102], which require modeling abstractions and mapping the arithmetic logic between language and math symbols [72, 200].

To tackle a manageable scope within this large domain, and since our studies are concerned primarily with enterprise documents, our work will focus on the question answering task over table/text input. In this task, the input is a natural language question with a table and surrounding text, provided as context. The output is a program made up of arithmetic operators and operands (see Table 1.2).

2.1.1 QA over tabular data

As with many modern QA models, most tabular QA approaches use a retriever-generator architecture [69], where the retriever identifies relevant table cells and encodes them using spatially-aware tabular encoding [47, 211] or verbalization [26]. The generator produces the program necessary to derive the answer. This provides the opportunity to measure model performance in terms of program accuracy as well as execution accuracy.

Numerous studies have tackled quantitative reasoning in retriever-generator models. Retriever-focused studies have proposed structure and number aware representations that model the magnitude, polarity, or relationship among quantities [194, 200]. The need for large-scale in-domain datasets limits the applicability of these methods. Hence, generator-focused studies have attempted to enhance quantitative reasoning at generation time, using graph-based reasoning [148, 220], knowledge infusion [120], logical programming [84], and causal reasoning [94]. In the latter, **counterfactual scenarios** are used to augment the data in such a way that the model generalizes to never-before-seen operands.

Despite major improvements, quantitative reasoning remains a challenge [4]. The challenge stems from the memorization of spurious lexical patterns by the model, especially in the absence of large-scale training data [149]. This is reminiscent of the problem of compositional generalization, which has been studied in-depth in numerous NLU fields including semantic parsing [40], visual question answering [156], data-to-text generation [116], and learning from instruction [98].

2.1.2 Compositional generalization

Compositional generalization is a model's ability to recognize new structures that are novel, but made up of previously seen components [119]. Oren et al. [125] explore several methods to improve compositional generalization for semantic parsing tasks, including the downsampling of repetitive patterns, using grammar-based decoding, and supervising the attention weights to ensure proper alignments are maintained between input and output terms. A method that consistently outperforms other approaches in text-to-SQL and tabular QA tasks is **attention coverage**. Coverage is a penalty term that encourages the model not to pay too much attention to familiar (i.e. frequently seen) terms and focus its attention weights on new, unseen terms at test time.

Yin et al. [212] propose a simple yet effective method to supervise attention weights for a semantic parser using a small number of samples. They first find span-level **alignments** between the natural language input and the program output using a heuristic algorithm. Next, they encourage attention weights to follow the alignments by adding a supervised attention loss. The loss can be thought of as a regularization term that prevents the model from overfitting to spurious patterns.

Inspired by the above studies, in Chapter 3 we propose two novel mechanisms for grounding quantitative reasoning in input expressions. Our methods not only outperform SotA models, but also improve the robustness and OOD performance across several distributions.

2.2 Grounded spatio-visual reasoning

The spatial signal (layout) and visual signal (design) of a multimodal document are often closely intertwined and inform each other [75]. Nevertheless it is common practice in the VRDU literature to segregate these signals and develop strategies that optimize each of them separately. In this section, we review these strategies and their relation to multimodal groundedness.

2.2.1 Exploiting the spatial signal (layout) of documents

The placement of textual elements on a page serves as an important semantic signal: Proximity between elements can indicate continuity or segmentation. Horizontal or vertical alignments within tables or forms can indicate correspondence.

Certain structures such as tables and bullet lists provide explicit spatial alignments that can be exploited to enrich representations of text. Other structures such as form-like layouts are not as explicit and require representations that are more flexible.

End-to-end approaches for capturing spatial information from tables are often inspired by the sequence modeling paradigm used in neural language models. Herzig et al. [47] propose TAPAS, a table-representation method that can be used for question-answering. TAPAS treats each table as a sentence, serialized by concatenating the cells in a row-wise fashion, where the header of the table forms the first row. It then augments each cell with several spatial features including positional encoding [183], column ids, row ids, and rank ids. The rank id refers to the ranking of the raw value of the cell compared to other cells in the table. This allows for `max` and `min` operations to be performed on the content.

This encoding enables the seamless integration of tabular content with textual content. As an example, it allows a question to be preprended to the table using a special `SEP` token. However, as Shaw et al. [161] have shown, despite the early success of additive positional encoding in capturing 1-D positions, 2-D positional encoding using horizontal and vertical coordinates (similar to the row and column ids used in TAPAS), is not very effective in capturing relative positions in two dimensions.

TABERT remedies this by adding a column-wise component that is the result of mean-pooling cell representations from the same column. They also add a vertical self-attention layer over this new component that can capture column-wise relationships across multiple rows. Using **cell recovery** and **masked column prediction** objectives, they are able to encourage the model to recover masked content in both direction. This effectively "grounds the objective(s)" of the model, as described in Table 2.1. TABERT outperforms previous state of the art models on question answering over tabular data.

As previously mentioned, tables are not the only components in a document that carry spatial information. Bullet points, borderless forms, and even spatial separation between paragraphs can carry important semantic signals. In order to encode these open-ended structures in a sequence model, the model needs to decide on the order by which the text components are serialized. A simple left-to-right and top-to-bottom order can break in case of multi-column pages or tables. Instead of forcing this order, studies such as ROPE [87], FormNet [88], and FormNetV2 [89] treat reading order as a walk on a $beta$-skeleton graph [79] over the tokens on the page. ERNIE-Layout [131] uses an explicit serialization module, which first breaks each page down into its major layout elements (e.g. paragraphs, tables, figures, lists), and then uses this information to adjust the ordering of elements if necessary.

To avoid having to serialize a page into a 1D structure, Xu et al. [205] propose a different strategy for their LayoutLMv2 model. Inspired by Raffel et al. [144], they propose a **Spatial-Aware Self-Attention** mechanism for their LayoutLMv2 model. The method can work with any input as long as a sequence of tokens and corresponding bounding boxes is supplied (such as the output of standard Optical Character Recognition tools). Given $\mathbf{x}_i$ and $\mathbf{x}_j$ as the query and key tokens for the ith and jth tokens on a given page, they calculate an attention weight α_{ij} using scaled dot-product attention [183].

$$\alpha_{ij} = \frac{1}{\sqrt{d_{\text{head}}}}(\mathbf{x}_i\mathbf{W}^Q)(\mathbf{x}_j\mathbf{W}^K)^\intercal \tag{2.1}$$

where d_{head} is the hidden size, $\mathbf{W}^Q$ is the query projection matrix and $\mathbf{W}^K$ is the key projection matrix. In a typical self-attention mechanism, α_{ij} would then be subject to a softmax function, but prior to doing that, the Spatial-Aware Self-Attention head augments it using the following bias terms:

$$\alpha'_{ij} = \alpha_{ij} + \mathbf{b}^{\text{1D}}_{i-1} + \mathbf{b}^{\text{2D}_\text{x}}_{x_j - x_i} + \mathbf{b}^{\text{2D}_\text{y}}_{y_j - y_i} \tag{2.2}$$

$\mathbf{b}^{\text{1D}}_{i-1}$ is a vector that is indexed according to the difference between the 1D positions of the two tokens. $\mathbf{b}^{\text{2D}_\text{x}}_{x_j - x_i}$ is a vector that is indexed by the horizontal difference between the two tokens. Similarly, $\mathbf{b}^{\text{2D}_\text{y}}_{y_j - y_i}$ is a vector that is indexed by the vertical difference between the two tokens. The horizontal and vertical differences are both grouped into 50 buckets, meaning both $\mathbf{b}^{\text{2D}_\text{x}}_{x_j - x_i}$ and $\mathbf{b}^{\text{2D}_\text{y}}_{y_j - y_i}$ are vectors of size 50. These additional bias terms allow the model to learn different attention weights depending on the pair-wise distances between the ith and jth token.

To obtain the final output representation $\mathbf{h}_i$:

$$\mathbf{h}_i = \sum_j \frac{\exp \alpha'_{ij}}{\sum_k \alpha'_{ik}} \mathbf{x}_j \mathbf{W}^V \tag{2.3}$$

where $\mathbf{W}^V$ is the value projection matrix. A similar approach is adopted by Hong et al. [51], but instead of using fixed buckets, they apply a sinusoidal function to the relative positions of the two tokens and apply additional parameters for the relative positions of each coordinate.

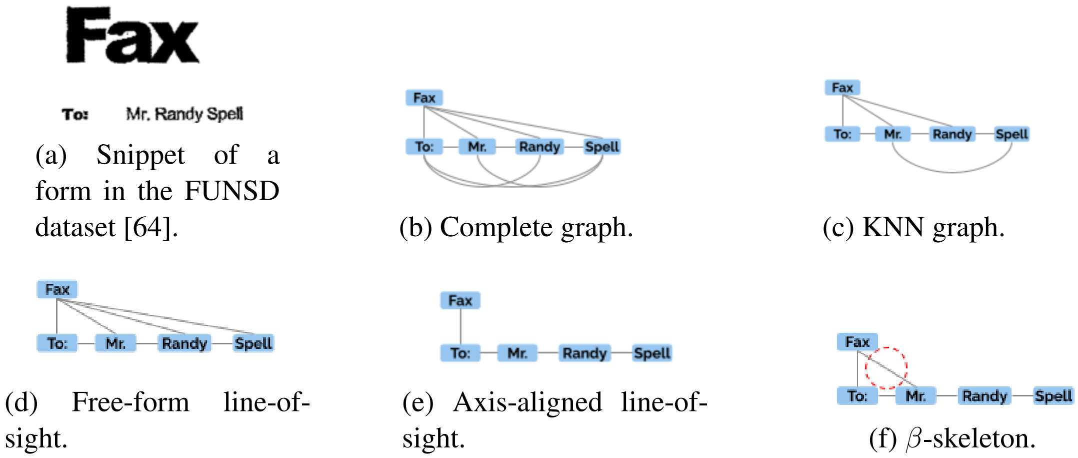
The Spatial-Aware Self-Attention mechanism allows the model to capture spatial relations in any context, including paragraphs and non-tabular segments. It is a way to “ground the design” of the model in the spatial structure of the document, as mentioned in Table 2.1. The resulting model outperforms state of the art on several tasks, including information and relation extraction from forms, document classification, and visual question answering over documents.

Despite the success of sophisticated spatial representations, studies such as DocLLM [189] have demonstrated that a simpler approach can perform well if the model is allowed to learn disentangled representations. Modeling the position of each token as the four coordinates of its bounding box, the model learns spatial representations by applying self-attention to the spatial modality alone, and later fuses it with text embeddings. This allows the model to learn non-linear relationships across modalities that can capture settings such as: `if token=`“Date” `AND` `position=left-top` AND `size=large,` then `class=HEADER`. In a way, the model disentangles or "grounds the search space", as described in Table 2.1.

Another approach to representing a document as a 2D structure is to use a graph-based representation. Graph structures allow more flexibility in representing information and controlling the way the information flows through the model. They can also be used seamlessly for several tasks, including document classification, semantic labeling and key information extraction, segmentation and relationship prediction, and document structure identification. Nevertheless, they remain under-explored in state of the art VRDU models.

A common method is to represent each token (or less commonly, each segment) as a node in a graph. The node can be represented using various information about the token, such as its text embedding, positional information, visual features, and other key characteristics.

A key aspect of graph design is the heuristic used for connecting nodes via edges. Figure 2.1 shows five possible ways to construct a graph, where some are more common than others.



(a) Snippet of a form in the FUNSD dataset [64].

(b) Complete graph.

(c) KNN graph.

(d) Free-form line-of-sight.

(e) Axis-aligned line-of-sight.

(f) β-skeleton.

Figure 2.1: A snippet of a form and five different ways to represent its contents as a graph.

- A **fully connected graph** would require each token to be connected to every other token on the same page. This dense representation is not used in practice, because of intensive memory and runtime requirements.
- A K**-Nearest Neighbor** heuristic can be used to connect each token to its nearest neighbors [139]. While this can alleviate the memory and runtime issues of a fully connected graph, it does not provide useful layout information to the model and is sensitive to the choice of K.
- A **line-of-sight** method improves on the KNN heuristic by connecting two tokens if they are in each other's line of sight, meaning there are no intermediary tokens between them. While this can be a useful strategy to model placement and spacing, it can be brittle [31]. Furthermore, the number of edges within a local context with n nodes remains quadratic, i.e. $O(n^2)$ [191].
- An **axis-aligned line-of-sight** method tries to improve on the line-of-sight heuristic by accounting for the grid-like structure of a page. Two nodes are connected if there are

no intermediary nodes between them *and* the nodes are horizontally or vertically aligned. This reduces the number of nodes but can lead to over-pruning, especially when two nodes aren't perfectly aligned [191].

- A β**-skeleton graph** tries to balance the benefits of a free-form line-of-sight graph with that of an axis-aligned line-of-site graph. Using a "ball-of-sight" strategy, it is less dense than the former, but still captures certain connections that the latter misses. This strategy is used in ROPE [87] as well as FormNet [88].
- Lastly, certain models such as Visual FUDGE use a neural module to predict whether an edge should exist or not [31].

Graph representations allow the model to be more parameter efficient, but in terms of performance on complex VRDU tasks, they still lag behind the much larger sequence-based models. This is demonstrated in the third segment of Table 2.2, where the performance of graph-based models is compared against sequence-based models. The fourth segment of the table, showing models that combine the benefits of graph-based structured with sequence models, show the strongest performance across key VRDU tasks.

In Section 4.2, we propose a novel graph-based architecture that matches or exceeds SotA models on multiple form understanding tasks, with less than 30% of the number of parameters. The parameter and data efficiency of our proposed approach results from the topological structure of the graph, which is deeply grounded in the layout of each document.

2.2.2 Modeling the visual signal (design) of documents

Recent research in the field of VRDU has largely been inspired by Vision-Language Models (VLMs), originally developed to tackle tasks such as image captioning [7] and retrieval [74]. Many VRDU models follow a process similar to open-domain image understanding models. They use a visual feature extractor such as a CNN [9], an image encoder such as U-Net or CLIP [136, 142, 153] or a Region Proposal Network (RPN) such as Mask-RCNN [46] or Faster-RCNN [151] to identify segments within each page, and capture common visual features for each segment [95, 205, 207].

Once visual features are extracted, they can be combined with text embeddings. Following the success of Transformer-based architectures [183], modern VLMs often incorporate them as a key component of fusion between the textual and visual signal [9, 50, 95, 207]. In encoder-based models, the visual encoder is paired with a text encoder, and trained on a multimodal task that is inspired by Masked Language Modeling [34]. In decoder-based models, the visual encoder is paired with a text encoder/decoder, and trained on an autoregressive task similar to next token prediction [141, 183].

Similar to spatial grounding, visual grounding can be encouraged through the three major approaches listed in Table 2.1. Certain studies such as Arctic-TILT [14] and UDOP [171] design modality-aware attention heads that are able to tie the textual representations to corresponding visual features. Others, inspired by the Masked Language Modeling objective [34], employ multimodal objectives such as Masked Vision-Language Modeling [9, 207] or Learning-To-Reconstruct [9, 131]. Lastly, some studies such as TILT [136] focus on augmenting the model's search space by synthesizing and perturbing input samples.

The top two segments of Table 2.2 show the performance of SotA encoder-based models and decoder-based models on four common VRDU tasks. The table demonstrates a few key trends among these models: First, while decoder-based models generally outperform encoder-based models on Visual Question Answering, they lag behind on Key Information Extraction. This might come across as counter-intuitive, because VQA is often considered a more challenging task in terms of multimodal reasoning. In Chapter 5 we will suggest that this might partly be due to how VQA performance is evaluated. Specifically, we demonstrate how common VQA evaluation metrics do not account for the groundedness of VRDU models, leading to scores that reward hallucinations.

Model	**# Params**	**Archi tecture**	**Doc Class.**	**Key Information Extraction**				**Relation Extraction**	**Question Answering**
			RVL-CDIP [44] (Accuracy)	**FUNSD [64] (F1)**	**CORD (F1)**	**SROIE [57] (F1)**	**Kleister-NDA [167] (F1)**	**FUNSD [64] (F1)**	**DocVQA [113] (ANLS)**
LayoutLMLARGE [207]	343M	Seq-Enc	94.43	0.7895	0.9493	0.9524	0.8340		0.7259
BROS [51]	138M	Seq-Enc		0.8121	0.9536	0.9548		0.6696	
SelfDoc [95]	137M	Seq-Enc	93.81	0.8336					
LayoutLMv2LARGE [205]	426M	Seq-Enc	95.64	0.8420	0.9601	0.9781	0.8520		0.8529
DocFormerLARGE [9]	536M	Seq-Enc	95.50	0.8455	0.9699		**0.8580**		
LayoutLMv3LARGE [56]	368M	Seq-Enc	95.93	0.9208	0.9746			0.8035	0.8337
Donut [73]	143 - 176M	Seq-Dec	95.30		0.8410				0.6750
Dessurt [29]	127M	Seq-Dec	63.20	0.6500				0.4230	0.9360
UDOP [171]	794M	Seq-Dec	**96.00**	0.9162	0.9758				0.8470
TILTLARGE [136]	780M	Seq-Dec	95.52		0.9633	0.9810			0.8705
Arctic-TILT [14]	<1B	Seq-Dec				0.9430			0.9020
DocLLM [189]	7B	Seq-Dec	91.80	0.5180	0.6740	0.9190	0.6030		0.6950
SMoLA-PaLI-X [198]	48B	Seq-Dec							0.9055
InternVL 1.5 [24]	76B	Seq-Dec							0.9090
InternVL 2 Pro [24]	40B	Seq-Dec							0.9506
Qwen-VL-Max [11]	unknown	Seq-Dec							0.9307
Qwen2-VL-Max [190]	72B	Seq-Dec							**0.9670**
Visual FUDGE [31]	17M	Graph-Enc		0.6652				0.5662	
ROPE [87]	unknown	Graph-Enc		0.5722					
FormNet [88]	217 - 345M	Graph-Enc		0.8469	0.9728				
FormNetV2 [89]	204M	Graph-Enc		0.9251	0.9770	0.9831			
DocGraphLMBASE [188]	unknown	Hybrid		0.8877	0.9693				0.6984
GraphLayoutLMLARGE [96]	372M	Hybrid		**0.9439**	0.9775				
GeoLayoutLM [110]	399M	Hybrid		0.9286	**0.9797**	**0.9870**		**0.8945**	

Table 2.2: The performance of various multimodal models on several VRDU tasks. The size of each model has been specified in terms of millions of parameters. Note that for some models, the size can change depending on the number of parameters required to train on a particular task or dataset. The Serialization column indicates how information on a page is serialized. Under each task, the datasets and the corresponding performance metrics have been specified.

Second, bigger model size (in terms of larger number of parameters) is not always associated with better performance across VRDU tasks. This can similarly be attributed to the importance of groundedness. Grounding the visual signal in VRDU models can address key challenges related to parameter and data efficiency. Certain studies have demonstrated that re-fashioning the vision encoder to adapt to visual features in documents can lead to parameter-efficient representations. For example, in LayoutLMv3 [56], the authors show that replacing LayoutLMv2's sophisticated RPN (pre-trained on open-domain images) with a linear embedding of pixel-level color features (trained on documents in an end-to-end fashion) can save close to 15% of the parameters without performance loss compared to LayoutLMv2. Similarly, in TILT [136] the authors show that a U-Net encoder [153] trained on documents as part of end-to-end training can outperform LayoutLMv2. They also propose a data augmentation strategy where affine transformations are used to change the position, angle, size, and shear of various visual elements.

In Arctic-TILT [14], the authors further enhance TILT's visual representations by proposing "fusion by tensor product" as an alternative to the then-common additive fusion mechanism between image and text embeddings. Inspired by Schlag et al. [157], the authors argue that the tensor product mechanism is more efficient at representing contextual relationships between text and image embeddings. This is demonstrated by Arctic-TILT's robust performance across several VRDU tasks, matching or surpassing models that are 70 times larger, using just below 1B parameters (see Table 2.2).

Models such as UDOP [171] and FormNetv2 [89] further developed the concept of contextual image embeddings by tying them directly to corresponding text embeddings. Instead of producing patch-level embeddings or using an image encoder that produces arbitrary visual tokens, they specifically develop visual tokens that correspond to the regions occupied by each token. This allows these models to "align" each token embedding with its corresponding vision embedding, leading to enhanced performance on tasks such as KIE (see Table 2.2).

In Section 4.1 we will extend the grounding ideas presented in this Section to propose a more efficient VRDU model for visually rich forms. By factoring the visual features into "patterns" (or clusters), we enable the model to capture key design features such as contrast and color templates.

2.3 Grounded evaluation

When evaluating model performance on discriminative tasks such as Classification or IE, the VRDU literature has largely followed the tradition of unimodal NLP, using standard metrics such as F1 or MAP [88, 207]. For generative tasks such as Summarization or VQA, the field has had to innovate. When benchmarks such as DocVQA [113] were first established, native vision-language multimodality was not common in VRDU models. This meant that the models relied on Optical Character Recognition software to detect the text on each page. In order to avoid over-penalizing the models for errors made by OCR engines, Mathew et al. [113] proposed Average Normalized Levenshtein distance (ANLS), which has since become the de facto metric used in most Document VQA benchmarks [114, 175, 176, 179]. ANLS focuses on the surface similarity between the ground truth and predicted answers and tolerates small errors. For example, if the ground truth answer is "`Apple`" and the predicted answer is "`App1e`" where the letter "`l`" is replaced by the digit "`1`", the ANLS score remains high whereas an exact match score would fail the model. This is to allow room for errors in optical character recognition.

One disadvantage of this approach is the fact that all misspellings are treated equally. For example, consider the difference between numbers "`1700`" and "`1788`". While the previous example could be considered a minor misspelling, misrecognition of digits can alter the value of numbers. Furthermore, the recent popularity of large vision-language models has introduced native multimodality into the VRDU field, and has thus decreased reliance on OCR software. This has encouraged research into alternatives to the standard ANLS metric. As an example, Peer et al. [130] have proposed ANLS*, a semantically grounded metric that accounts for the semantic category of the ground truth and predicted answers.

In Chapter 5 we introduce a new evaluation framework for Document VQA models. Our framework accounts for quantitative as well as spatio-visual groundedness of the model's generations, and is configurable according to the end-user's requirements. We show that our proposed framework is better correlated with robustness and calibration for modern VRDU models.

2.4 Takeaways

The studies covered so far have all exploited various characteristics of spatial, visual, and quantitative signal in documents. The main distinction is in the way such grounding is encouraged throughout training. Given the current literature, three such approaches can be identified:

- The neural *design* of a model can be configured to ground its reasoning in certain signals within the input.
- The training *objective(s)* can be designed in such a way that they encourage model parameters to capture certain multimodal signals.
- Grounding can be enforced directly on the *search space* of the problem that the model is attempting to solve. This is possible by imposing explicit rules or constraints on the output, pruning the search space in a deliberate fashion, or curating the data to guide the model throughout training.

Table 2.3 organizes the studies presented in this section into these three approaches. While there has been considerable effort in each category, there remain a few gaps in the way multimodal groundedness is encourage in SotA models.

- First, objective-based approaches are underexplored in the quantitative reasoning space. In spatial and visual reasoning, self-supervised objectives such as masked column prediction and masked visual language modeling are used extensively to encourage models to create expressive representations for document layout. In contrast, the attention alignment objective used in quantitative reasoning is supervised and requires the creation of explicit labels that map components in the input to those in the output. Attention coverage is an unsupervised objective, but it does not leverage the power of self-supervision to adapt itself to different contexts.
- Second, graph-based representations are used to model spatial relationships and the relationship among various quantitative concepts. However, they have not been explored in the context of visual reasoning. Similarly in adjacent domains such as open-domain VQA, scene graphs are often used to model spatial or relational aspects rather than visual contrast between objects [58]. In documents, the contrast in visual features can be just as important in navigating a document's layout as the contrast in spatial placement, indicating correspondence, importance, and segmentation.
- Third, approaches that explicitly scaffold the search space are quite rare. Researchers often prefer to curate the space by augmenting their datasets and avoiding the addition of any explicit constraints. This could be due to the brittle-nature of rules based approaches. However, Ravichander et al. [149] have shown how proper scaffolding can be used to improve numerical reasoning in quantitative NLI tasks. For example, QA tasks can prune

the space of answers that are impossible for a given context. This area calls for further exploration.

We will attempt to address the above gaps in current research in multimodal document understanding, aiming to conclude with a holistic approach to model evaluation that can account for spatial, visual, and quantitative reasoning. The following chapters will cover our contributions to each of the above domains.

Reasoning	Grounding the design	Grounding the objective	Grounding the search space
Quantitative	Graph-based representation [148, 220]	Attention coverage [125] Supervised alignments [212]	Data augmentation via counterfactuals [94]
Spatio-visual	Spatial-aware attention [205] Graph-based representation [31, 87, 88] Modality-adaptive attention [95]	Cell recovery [47, 211] Masked column prediction [211] MVLM [9, 207] TIA/TIM [9, 56, 205] LTR [9, 131]	Probabilistic soft logic [169] Data augmentation via perturbations [136]

Table 2.3: Example studies that have examined various approaches to grounding the meta-textual signal in document understanding.

Chapter 3

Grounded learning objectives for quantitative reasoning

In Section 1.1, we introduced Alice, a knowledge worker at a financial institute who is tasked with processing authorized signatory forms. These forms are often composed of a list of authorized signatories, their titles, and contact information. In addition, each signatory might have limited authorization with regard to different types of transactions. For example, they might only be authorized to sign off on transactions between \$1 million and \$10 million in value. These limitations can be expressed in natural language (e.g. "between 1MM USD and 10MM USD") or using math symbols (e.g. "\$1MM $<$ and $<$ \$10MM"). Processing these limitations into a standard canonical form would require some understanding of mathematical operations.

This is a relatively simple case of quantitative reasoning that is required in VRDU models. Processing financial disclosures, loan documents, analytical reports, and other forms of enterprise documents would require more complex reasoning over quantitative data interspersed with natural language expressions. As an example, consider the task of calculating the average revenue of a given company over a period of three quarters, based on their latest financial report. The task would require two major steps:

1. **Retrieving** the revenue for each year from the document.
2. **Generating** the response by performing multiple operations: adding the revenue of each year and dividing by 3.

Many modern Quantitative Question Answering models follow the above steps in a framework known as the **Retriever-Generator** architecture [69]. A key challenge of this architecture is that the generator can suffer from overfitting to spurious patterns, especially when it needs to generate multi-step operations [26]. As an example, if the token "2019" repeatedly appears in samples that require a `division` operation, the generator might produce `division` whenever it encounters "2019". As we will describe in Section 3.1.4, this problem is related to *Compositional Generalization*, which was first introduced in Section 2.1.2.

In this chapter, we explore how improving the groundedness of Quantitative QA models can alleviate the challenge of Compositional Generalization. Toward that goal, we propose two new methodologies for multi-step quantitative reasoning that ground the reasoning process in specific expressions within the input, leading to more robust models. The previous chapter concluded with a list of common approaches by which groundedness is encouraged in state of the art models. Table 3.1 includes the same summary, but with three additional entries (highlighted in blue). These entries denote our contributions covered in this chapter.

Reasoning	Grounding the design	Grounding the objective	Grounding the search space
Quantitative	Graph-based representations [148, 220]	Attention coverage [125] Supervised alignments [212] Unsupervised alignments (Section 3.1) Metric learning (Section 3.2)	Data augmentation via counterfactuals [94] Counterfactual sampling (Section 3.2)
Spatio-visual	Spatial-aware attention [205] Graph-based representation [31, 87, 88] Modality-adaptive attention [95]	Cell recovery [47, 211] Masked column prediction [211] MVLM [9, 207] TIA/TIM [9, 56, 205] LTR [9, 131]	Probabilistic soft logic [169] Data augmentation via perturbations [136]

Table 3.1: An updated view of Table 2.3, where our proposed methods have been added to corresponding cells, blue. Each highlight has a reference to the section where it is covered.

In Section 3.1, we will demonstrate how unsupervised objectives can be used in QA models to improve compositional generalization for quantitative reasoning. In Section 3.2, we will lay out a method that uses counterfactual scenarios to sample negative and positive samples in such a way that metric learning can lead to better compositional generalization for quantitative reasoning.

3.1 Operator-aware attention

Quantitative reasoning is an important aspect of question answering, especially when numeric and verbal cues interact to indicate sophisticated, multi-step programs. In this section, we demonstrate how modeling the compositional nature of quantitative text can enhance the performance and robustness of QA models, allowing them to capture arithmetic logic that is expressed verbally. Borrowing from the literature on semantic parsing, we propose a method that encourages the QA models to adjust their attention patterns and capture input/output alignments that are meaningful to the reasoning task. We show how this strategy improves program accuracy and renders the models more robust against overfitting as the number of reasoning steps grows. Our approach is designed as a standalone module which can be pre-pended to many existing models and trained in an end-to-end fashion without the need for additional supervisory signal. As part of this exercise, we also create a unified dataset building on four previously released numerical QA datasets over tabular data.

3.1.1 Background

Any natural language system that processes or interacts with numeric data requires quantitative reasoning to function. This has inspired research in several NLP domains, including reading comprehension [8, 118], textual entailment [149, 154], data-to-text generation [127, 168], and question answering [22, 220]. A major challenge in quantitative reasoning is the interplay between numeric expressions and natural language [154]. Standard neural approaches rely heavily on lexical matching, leading to overfitting over spurious verbal patterns. In contrast, a purely symbolic approach excels at numerical reasoning, but struggles when sophisticated verbal reasoning is required [149]. We introduce a novel attention strategy that captures the interplay between numeric and verbal modalities, which improves program accuracy and renders models more robust to overfitting.

Focusing on the question answering task, we show how our proposed method, named **CompAQT** (COMPositional Attention for QuanTitative reasoning), enables the model to attend to relevant parts of text at each reasoning step. CompAQT enhances the performance of state of the art models on several recently released QA datasets, especially for multi-step programs. It is implemented as a plug-and-play module that can be added to existing models with minimal effort and without the need for any additional supervision.

Concretely, we offer the following contributions:

- We propose a compositional attention module equipped with an alignment loss that improves SOTA performance on numeric QA tasks.
- We demonstrate how the proposed approach improves the models' program accuracy and renders them more robust in multi-step reasoning tasks.
- We combine and refine four recently released datasets on QA over tabular data. We unify their annotation schema so that they can be used interchangeably.

3.1.2 Problem statement

In the retriever-generator configuration of a QA model, our goal is to improve quantitative reasoning in the generator component. Figure 3.1 illustrates the typical architecture of a generator as an encoder-decoder model. The encoder uses a contextual representation model such as RoBERTa [107]. The decoder combines a recurrent module with one or more cross-attention heads between the natural language input and the program output. As the output is generated step by step, it is crucial for the cross-attention module(s) to capture relevant components of the input, otherwise they can simply memorize spurious verbal patterns and fail to generalize, especially as the number of steps grows in the output.

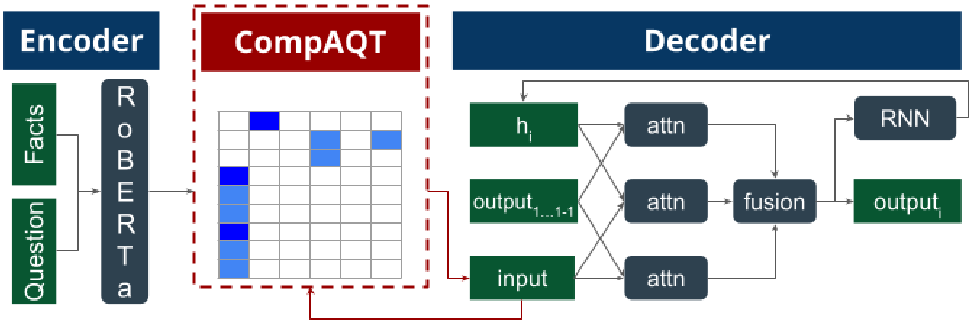


Figure 3.1: The typical encoder-decoder architecture of a quantitative QA generator. We introduce the compositional attention component (middle, enclosed in dotted line) to enhance the alignments between natural language input and program output.

Table 3.2 illustrates this phenomenon with four examples from a QA task. Each row displays a natural language question, the set of facts that can be used as evidence to answer the question, and the program to arrive at the correct answer. Presented with the first three examples, it is conceivable that a human would be able to extrapolate that "percent change" is calculated by first measuring the net change (i.e. subtraction) and then scaling the number as a percentage. Humans are able to do this by recognizing components in the question that have been previously encountered (e.g. "percent change" and "expenses") even if they were not encountered in this particular arrangement.

	Question	Evidence	Program
1	What was the net change in ***revenue*** from ***2019*** to ***2020***?	***2019 revenue*** was $*80*M ***2020 revenue*** was $*60*M	`subtract`(*80*, *60*)
2	What was the net change in ***expenses*** between ***2018*** and ***2021***?	***2018 expenses*** were $*20*M ***2021 expenses*** were $*30*M	`subtract`(*20*, *30*)
3	What was the percent change in ***revenue*** from ***2019*** to ***2020***?	***2019 revenue*** was $*80*M ***2020 revenue*** was $*60*M	`subtract`(*80*, *60*) `divide`(#0, *80*) `multiply`(#1, 100)
4	What was the percent change in ***expenses*** between ***2018*** and ***2021***?	***2018 expenses*** were $*20*M ***2021 expenses*** were $*30*M	`subtract`(*20*, *30*) `divide`(#0, *20*) `multiply`(#1, 100)

Table 3.2: Example of compositional alignments between input questions and output programs in the financial QA task. Blue underlined text indicates terms that relate to arithmetic operators. *Red italicized* text indicates terms that relate to operands. ***Bold italicized text*** indicates terms that are shared between the question and evidence.

Many neural models struggle to exhibit the same behavior, due to overfitting to spurious patterns in natural language, or in the output. As we will later discuss in Section 3.1.4, quantitative reasoning datasets can exhibit a long-tail distribution, biased towards simpler patterns. Figure 3.2 illustrates how this phenomenon takes place in the training split of one such dataset. The figure shows the prominence of the most common sequences of arithmetic operators in the FinQA training set [26]. As the number of steps grows, the tail grows longer and the sample size smaller, thus providing less information to the model and forcing it to rely on repetitively encountered patterns in the past.

Our goal is to encourage the model to focus its attention on relevant components of the input during generation. Figure 3.3 illustrates the expected attention patterns for the fourth example from Table 3.2. The figure illustrates two key points: 1) During program generation, the terms that overlap between the question and the evidence do not matter as much as non-overlapping terms. 2) When generating operators (such as `subtract` or `divide`) attention should be focused on terms that are exclusive to the question. Whereas when generating the operands (such as `80` or `20`), attention should be focused on terms that are exclusive to the evidence. Constants such as `100` or `#0` may depend on the question, the facts, or the previously generated steps.

Using this insight, we encourage the model to adjust its attention patterns accordingly. The remaining sections describe our methodology and experimental results in detail.

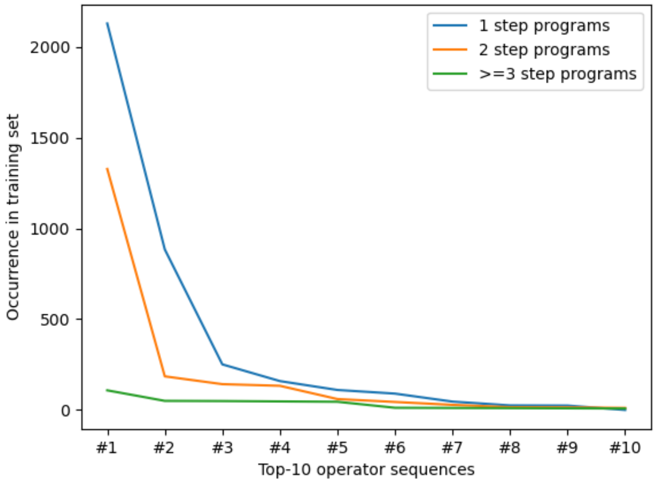


Figure 3.2: The long-tail effect in multi-step programs in the FinQA training set [26].

3.1.3 Methodology

Let Q be a question made up of a sequence of tokens $q_1, \cdots, q_n$. Let F be the evidence obtained by the retriever, made up of a sequence of tokens $f_1, \cdots, f_m$. Note that the evidence can be composed of one or more concatenated facts, as illustrated in Table 3.2. Consistent with [26], we represent the output program S as a sequence of steps $s_1, \cdots, s_l$. Each step s_i is composed of an operator o_i (such as `divide` or `subtract`) and exactly two operands, $a_{i,1}$ and $a_{i,2}$[1]. An operand can have one of three values: 1) It can be one of the tokens in F. 2) It can be a constant used in scaling or counting operations, e.g. `const_100`. The list of possible constants is pre-defined. 3) It can be a reference to a previous step, e.g. `#0`. The maximum number of steps is pre-defined.

Given a retriever-generator model, we pre-pend a self-attention module to the generator, as illustrated by the red dotted box in Figure 3.1. First, we encode $Q||F$ using a contextual embedding model such as RoBERTa [107] with embedding size d_{enc}. This results in an embedding matrix $\boldsymbol{U} \in \mathbb{R}^{d_{\text{enc}} \times (n+m)}$. At each generation step i, we apply scaled dot-product self-attention [182] to $\boldsymbol{U}$, resulting in the attention grid $\boldsymbol{A}^{(i)} \in \mathbb{R}^{(n+m)\times(n+m)}$ and the attention output $\boldsymbol{X}^{(i)} \in \mathbb{R}^{d_{\text{enc}} \times (n+m)}$. Our goal is to encourage $\boldsymbol{A}^{(i)}$ to focus its alignments properly, such that $\boldsymbol{X}^{(i)}$ supplies relevant information to the generator.

We follow a similar strategy to [212], but in the absence of gold alignments, use the heuristics described in Section 3.1.2. Concretely, we add the below term to the loss:

$$\mathcal{L}^{(i)}_{\text{align}} = \frac{1}{n+m} \sum_{k=1}^{n+m} \sum_{j=1}^{n+m} (a^{(i)}_{j,k} - p_{\text{prior}}(\boldsymbol{u}^{(i)}_j | \boldsymbol{u}^{(i)}_k))^2 \tag{3.1}$$

where $a^{(i)}_{j,k}$ is the attention weight between the jth and kth tokens in $\boldsymbol{A}^{(i)}$, and $p_{\text{prior}}(\boldsymbol{u}^{(i)}_j | \boldsymbol{u}^{(i)}_k)$

[1] We follow the notation used by FinQA, where programs are modeled as right-expanding binary trees with each operation having two operands. If necessary, one or more operands are set to `NONE`.

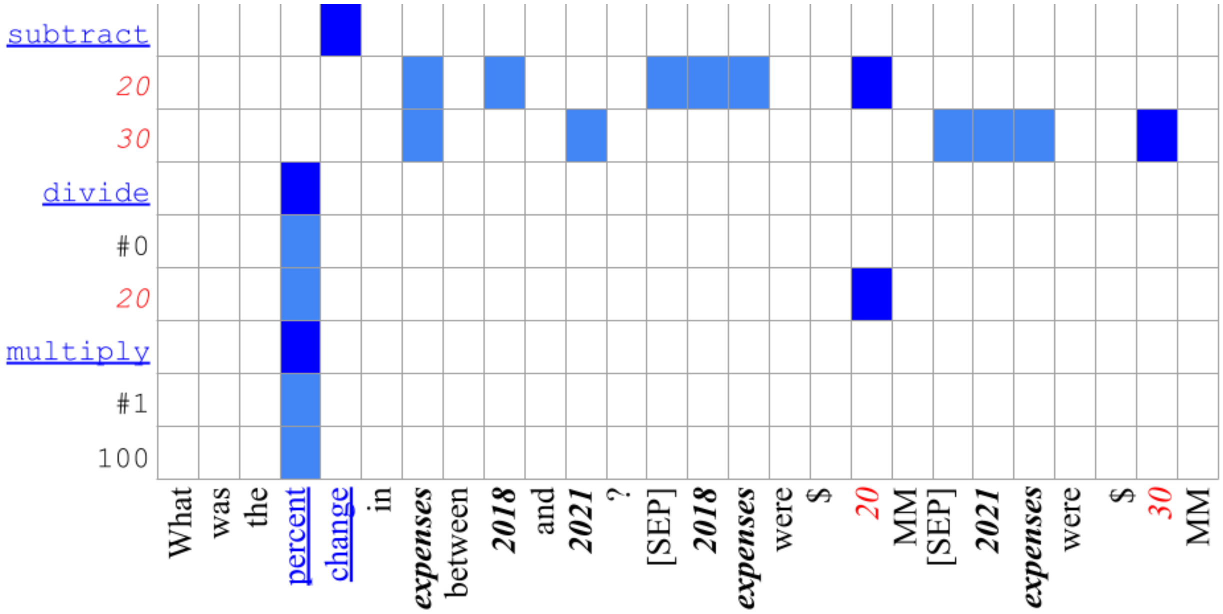


Figure 3.3: Semantic alignments between the natural language input from a financial QA task, and the corresponding program output.

is defined as:

$$\max\{0, \min_{j' \neq k}(\text{dist}(\boldsymbol{u}_{j'}^{(i)}, \boldsymbol{u}_k^{(i)})) - a_{j,k}^{(i-1)}\}$$

where dist is the cosine distance between two vectors, scaled between 0 and 1, and $a_{j,k}^{(0)} = 0$ for all j and k.

The term $\min_{j' \neq k}(\text{dist}(\boldsymbol{u}_{j'}^{(i)}, \boldsymbol{u}_k^{(i)}))$ encourages the model to distribute attention to each token based on its closest similarity to any other token in the input. This is balanced against the previous attention distribution $a_{j,k}^{(i-1)}$, leading to the following behavior:

1. For tokens that are repeated more than once (e.g. those tokens shared between the question and the evidence), lower attention is encouraged. This helps the model to disregard tokens such as “expenses” and “2019” illustrated in Figure 3.3.

2. For terms that are unique to the question or the evidence, high attention is encouraged in early steps. This helps the model to focus on tokens such as “percent” and “20”.

3. In later steps, the model is discouraged from focusing on previously well-attended tokens. For instance after the model attends to the word “change” in order to generate `subtract`, it learns to shift its focus away.

(1) and (2) emulate the regularization strategy proposed by [212], while (3) emulates the concept of coverage proposed by [125] with the contrast that it tracks tokens seen in previous generation steps for the same sample. The total alignment loss for a sample is calculated as an aggregation over all steps, with linear decay:

$$\mathcal{L}_{\text{align}} = \frac{1}{l} \sum_{i=1}^{l} \mathcal{L}_{\text{align}}^{(i)} - \alpha i \tag{3.2}$$

The linear decay term helps the model assign a higher penalty to earlier generation steps. Suppose the gold program is `subtract(20, 30), divide(#0, 20)`, and the output generated by the model is either `subtract(20, 30), multiply(#0, 20)` or `add(20, 30), multiply(#0, 20)`. In the absence of linear decay, both predictions would receive the same penalty. The decay term assigns a lower penalty to the first prediction, since it gets the first operation correct. Finally, the alignment loss can be added to the default loss of the generator:

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}_{\text{align}} + (1 - \lambda)\mathcal{L}_{\text{generator}} \tag{3.3}$$

3.1.4 Experiments

In this section, we describe our experimental set up, including the datasets and the baseline models.

Datasets

We use four datasets that focus on numerical reasoning over hybrid table/text context, all released within a year of this publication.

FinQA [26] is based on a collection of financial reports published by U.S. companies that were released as part of FinTabNet [226]. Each passage is composed of a table and a few sentences that surround the table, describing its content. The questions, designed by domain experts, all require numerical reasoning.

TAT-QA [227] is also focused on financial reports, but includes documents from non-U.S. companies. As such, the reports do not conform to a standard format and include a more diverse set of metrics. The dataset includes span-based and multi-span questions as well as questions requiring arithmetic reasoning. In our experiments, we focus on the latter category.

HiTab [27] is a collection of tables that include statistical data, collected from various national agencies. The tables have complex hierarchical or nested structure, and answering them requires spatial as well as numerical reasoning. As with TAT-QA, we discard questions that do not require any arithmetic operations.

MULTIHIERTT [224], which is also based on FinTabNet, combines the challenges of the above-mentioned datasets, bringing together complex tabular structures and hybrid table/text contexts. Again, we filter the dataset down to those samples that require numerical reasoning.

All four datasets provide the reasoning program required to derive the answers, allowing any model to be evaluated on program accuracy. Since our study is focused on multi-step generation, we use program accuracy as our evaluation metric.

Unified dataset Of the datasets mentioned above, FinQA is exclusively focused on multi-step quantitative reasoning. The remaining datasets tackle additional challenges such as extractive QA, spatial reasoning, and table representation. Therefore we filter TAT-QA, HiTab, and MULTIHIERTT down to those sample that require quantitative reasoning. We also transform each sample so that it conforms to the standard FinQA format. This helps us bypass the challenge of addressing complex tabular structures, which is out of scope for this study. Please refer to Appendix A.1 for further details. Table 3.3 shows statistics for each dataset, as well as the distribution of 1 step, 2 step, and 3+ step programs.

Dataset	# passages	# QA pairs used			# steps in program		
		Train	Dev	Test	1 step	2 steps	3+ steps
FinQA	2,789	6,251	883	1,147	4,894	2,709	678
TAT-QA[2]	2,479	4,355	230	307	2,721	616	1,555
HiTab	513	879	186	199	1,075	120	69
MultiHiertt[3]	2,291	2,083	100	108	560	862	869
Combined	8,071	13,568	1,349	1,761	9,520	4,388	3,171

Table 3.3: Statistics of the four datasets, and the combined dataset. Note that with the exception of FinQA, only a subset of samples (involving multi-step quantitative reasoning) is used from each dataset.

Baselines

To establish baselines, we use the following models, which have demonstrated SOTA performance on the datasets mentioned in the previous section[4].

FinQANet was proposed by [26] and applied to the FinQA dataset. The architecture is similar to that illustrated in Figure 3.1 but missing the compositional self-attention module.

TagOp was proposed by [227] and applied to the TAT-QA dataset. A crucial difference between TagOp and FinQANet is that the former is not designed to perform multi-step reasoning, but approaches the task as a classification problem. As an example, it may predict that a `change ratio` calculation is required, which implies a subtraction followed by a division. Seven such arithmetic operations are permitted.

In addition to the above, we use a pointer-verbalizer network (**PVN**) as a universal baseline against all four datasets. The model is inspired by the Expression-Pointer Transformer proposed by [72]. The authors argue that generating an arithmetic program as a disjoint sequence of operators and operands is not consistent with how humans approach quantitative reasoning. Instead, they propose the concept of an "Expression Token", which represents a full operation autonomously (e.g. instead of generating `divide`, `20`, `30` as a sequence, they recommend generating `divide(20, 30)` as one token). Following this idea, PVN also generates Expression Tokens, but uses two pointer mechanisms—one to select operators from the list of all possible options, and one to select operands from the list of numbers expressed in the evidence, or a predetermined list of possible constants. In addition, it uses verbalization to map operators from a symbolic space (e.g. `+`) into the semantic space (e.g. "divide"). Please refer to Appendix A.2 for implementation details.

We use the above three models as baselines, and measure their performance before and after adding CompAQT. To remain as consistent as possible with the initial settings of these mod-

[2] TAT-QA samples includes a flag to distinguish arithmetic questions from span-based questions. However, this flag is only available in the train and dev sets, but not in the test set. Therefore we split the dev set into 230 dev examples and 307 test examples.

[3] MultiHiertt does not include an annotated test set. Therefore we split the validation set into 100 dev examples and 108 test examples.

[4] The creators of HiTab and MultiHiertt have proposed baseline models that were not included in our experiments. This is because the HiTab model is focused on encoding tabular data rather than quantitative reasoning. The MultiHiertt model, named MT2Net, is similar to FinQANet, but includes an additional sub-module that only applies to span-based questions—again, out of the scope of this study.

els, we use the same hyperparameters and settings described in the original papers. We also use RoBERTa-large [107] to encode the input, since all models report best performance on this model. We perform grid search on the development set of FinQA to tune the values for α and λ, which are subsequently both set to 0.1. We use the same values throughout all of our experiments. Please refer to Appendix A.3 for additional details and the full list of hyperparameters for each baseline.

Note that TAGOP has been designed for single-step programs. Therefore we apply it to the original version of the TAT-QA dataset, but analyze the results based on the actual number of steps in each program. When adding CompAQT to TAGOP, we pre-pend it once, and instead of using the multi-step loss with linear decay, we only calculate the alignment loss once per program. Further, note that since our study is only focused on generation and not retrieval, we use gold facts provided by each dataset. Please refer to Appendix A.4 to see details of experiments using retrieved facts.

3.1.5 Results and discussion

In this section, we investigate the effectiveness of CompAQT through four questions: 1) Does CompAQT improve the performance of the baseline models on the four datasets? 2) Does CompAQT encourage compositional generalization by enabling the models to attend to relevant parts of the input? 3) Does each component of CompAQT contribute to enhanced performance? 4) Can CompAQT's performance be attributed merely to added parameters? Additionally, we examine whether the combined dataset offers an advantage over the largest constituent dataset.

Model performance

Table 3.4 shows the program accuracy of each baseline model, before and after adding CompAQT. As the table illustrates, CompAQT's significantly contributes to performance on multi-step programs in three of the four datasets. An interesting exception is the TAT-QA dataset, which does not exhibit the long-tail distribution displayed in Figure 3.2. As Table 3.3 shows, TAT-QA is biased towards 3-step programs with repeating patterns (e.g. `change ratio` is a common 3-step program). Here, CompAQT offers comparable performance to the baseline, with slightly higher robustness to multi-step programs, sometimes at a slight cost to single-step programs. For datasets that exhibit the long-tail distribution, CompAQT offers improvement on all categories, but especially on multi-step programs. This is especially noteworthy for HiTab, which is the smallest and most skewed collection.

Among the three baselines, FinQANet outperforms others on individual as well as the combined dataset. Adding CompAQT further improves FinQANet's performance on all datasets with the exception of TAT-QA. Therefore we use FinQANet+CompAQT for the remaining analyses presented in this section.

[5] Note that TAGOP cannot generate multi-step programs and can therefore only be applied to the TAT-QA dataset, where multi-step programs have been collapsed into single-step operations (e.g. `change ratio`).

Model	**Dataset**	**Program accuracy**			
		1 step	**2 steps**	**3+ steps**	**Overall**
TagOp[5]	TAT-QA	45.01	39.56	42.73	43.25
+CompAQT		+1.06	+0.72	+1.00	+0.63
PVN	Combined	68.14	61.33	13.54	56.64
+CompAQT		**+2.64**	**+2.12**	**+3.09**	**+2.57**
FinQANet	FinQA	75.63	65.87	30.36	68.44
+CompAQT		**+3.05**	**+9.25**	**+5.49**	**+5.30**
FinQANet	TAT-QA	73.33	63.76	64.88	70.71
+CompAQT		**-3.33**	+0.00	+1.38	-0.74
FinQANet	HiTab	34.70	25.14	15.91	30.12
+CompAQT		+0.03	**+4.50**	+1.44	**+2.11**
FinQANet	MultiHiertt	38.99	40.07	15.01	38.94
+CompAQT		-0.12	**+2.28**	+1.76	**+1.88**
FinQANet	Combined	65.11	62.00	30.76	58.60
+CompAQT		+1.49	**+3.14**	**+3.40**	**+2.28**

Table 3.4: Program accuracy for program generation using baseline models. Additional performance gain/loss is indicated after CompAQT is added to each model. **Bold** numbers indicate that a gain/loss is significant at $p < 0.005$, based on the paired-bootstrap test proposed by [12], with $b = 10^3$.

Qualitative examples

Table 3.5 shows four examples from the validation set of the FinQA dataset. The first question asks for a percentage calculation. Percentage calculations are the most common two-step operations in the training set, and the FinQANet model is able to produce the gold program without any additional guidance from CompAQT. In the second example, the model is asked to perform an operation on two metrics that are expressed as percentages. This time, possibly by relying heavily on memorizing the relationship between the word “percentage” and the `subtract-divide` operation, FinQANet mistakenly generates a `subtract-divide` sequence, whereas CompAQT is able to determine that “percentage” refers to an operand. In the third example, FinQANet once again performs a percentage calculation, possibly by associating the word “change” with “percentage change”. Once again CompAQT is able to drive attention towards the correct program, and distinguishes between a net and a percent change. The final example shows a case where CompAQT is not able to improve baseline performance. This challenge here is to understand the relationship between the number of shares and the average price. This requires a level of financial literacy that is not resolved by compositional generalization alone. As demonstrated in [26] financial expertise plays a major role, even in human performance.

Attention patterns

To confirm whether CompAQT is assisting the model in detecting key operational terms, we analyze the top-attended tokens within the input. Table 3.6 lists the top attended tokens throughout the training process for the FinQANet+CompAQT model on the FinQA dataset. As training progresses, CompAQT learns to attend to key terms that indicate arithmetic operations (such as

Question	Evidence	Gold program	FinQANet	FinQANet + CompAQT
What was the percentage change in the fair value from 2010 to 2011?	1) the fair value of 2011 is \$99 2) the fair value of 2010 is \$81	`subtract(99, 81), divide(#0, 81)`	`subtract(99, 81), divide(#0, 81)`	`subtract(99, 81), divide(#0, 81)`
What was the difference in operating profit as a percentage of net sales between 2001 and 2003?	1) the company reported operating profit as a percent of net sales 2) operating profit in 2001 is 19 3) operating profit in 2003 is 26	`subtract(26, 19)`	`subtract(26, 19), divide (#0, 19)`	`subtract(26, 19)`
What is the change in the warranty reserve from 2017 to 2018?	1) balance as of 2017 is \$23 2) balance as of 2018 is \$24	`subtract(24, 23)`	`subtract(24, 23), divide(#0, 23)`	`subtract(24, 23)`
For the 4th quarter of 2011 approximately how much was spent on stock repurchases?	1) total number of shares purchased is 3915 2) total of average price paid per share is \$98	`multiply(3915, 98)`	`add(3915, 98)`	`add(3915, 98)`

Table 3.5: Four examples from the FinQA dataset, showing CompAQT's success and failure in capturing compositional expressions. Note that some numbers have been truncated to save space.

"net" and "growth"). Using a basic self-attention module shows a similar convergence, but the module is not as quick to learn important terms. In fact even at the 50th epoch, the basic self-attention module is still focusing its attention on terms that do not indicate an operation, such as "annual" and "year". This shows that the additional components in CompAQT (alignment and coverage loss) assist the model in converging to more meaningful attention patterns.

	Top attended token		
	Epoch #1	**Epoch #25**	**Epoch #50**
Self-attention	[CLS], ?, what, and	company, what, year, 2018	percentage, ratio, annual, year
CompAQT	the, of, ?, what	year, company, percentage, annual	percentage, growth lowest, net

Table 3.6: Top attended tokens throughout the training process for a vanilla self-attention module versus CompAQT. As training progresses, CompAQT learns to attend to tokens closely associated with quantitative operations. The results are based on FinQANet+CompAQT, applied to the FinQA dataset.

Ablation study

We perform a series of experiments to examine the impact of each component of CompAQT. Table 3.7 shows the results after applying FinQANet to the FinQA dataset. "Self-attention" indicates the addition of a plain self-attention module without any compositional guidance. "Alignment loss" indicates the addition of the minimum-distance component in Equation 3.1.3. "Coverage term" indicates the addition of $-a_{j,k}^{(i-1)}$ to alignment loss. "Linear decay" indicates replacing a simple average loss with the linear decay term in Equation 3.2. As the table shows, each component contributes to the program accuracy. The self-attention module offers an improvement that is relatively consistent across all programs, whereas the alignment loss and the coverage term favor multi-step programs, as intended. Lastly, linear decay further improves results for the longest programs by a small margin.

To ensure that the effectiveness of CompAQT is not simply due to added parameters, we also perform a series of experiments that measure the performance of CompAQT with additional parameters in the form of additional layers and attention heads. Each row in Table 3.8 shows how much program accuracy improves over using FinQANet without CompAQT. As the table shows, additional parameters are not always helpful and can undermine the performance of the model, especially for multi-step programs. This may also indicate that the regularizing effect of CompAQT can be counteracted by larger parameters, leading to overfitting over smaller datasets.

Pre-training across datasets

Since the combined dataset uses the same format for all constituent datasets, it is easy to investigate the impact of pre-training on larger and more diverse data. Figure 3.4 shows how FinQANet+CompAQT performs on the combined test set, as more collections are added to its training set. As the Figure illustrates, despite providing a sizeable number of single-step programs, TAT-QA fails to improve the performance substantially on multi-step programs. This is likely due to the fact that all TAT-QA programs fall into seven categories, which allows the model to memorize them. In contrast, despite its small size, adding MULTIHIERTT improves performance on 1 step and 2 step programs. 3+ step programs remain a challenge across all datasets, but the model shows steady progress as the dataset size grows.

Model	**Program accuracy**			
	1 step	**2 steps**	**3+ steps**	**Overall**
FinQANet	75.63	65.87	30.36	68.44
+self-attention	+2.95	+2.08	+2.00	+2.57
+alignment loss	+0.07	+5.31	+2.01	+1.95
+coverage term	+0.03	+1.97	+0.97	+0.69
+linear decay	+0.00	+0.07	+0.51	+0.06

Table 3.7: Ablation results on the FinQA dataset using FinQANet as the base model.

# heads	**# layers**	**# params**	**# Program accuracy (improvement over baseline)**		
			1 step	**2 steps**	**3+ steps**
1	1	4.2M	+3.05	+9.25	+5.49
4	1	4.3M	+3.97	+7.30	+5.31
1	2	8.4M	+4.22	+6.76	+3.12
4	2	8.6M	+4.26	+5.99	+2.86

Table 3.8: The performance of FinQANet+CompAQT on the FinQA dataset. As additional parameters are added in the form of multiple heads or more layers, the model's performance does not increase.

3.1.6 Conclusion

In this study, we proposed a method to improve multi-step quantitative reasoning for question answering. Our method facilitates compositional generalization by encouraging the model to attend

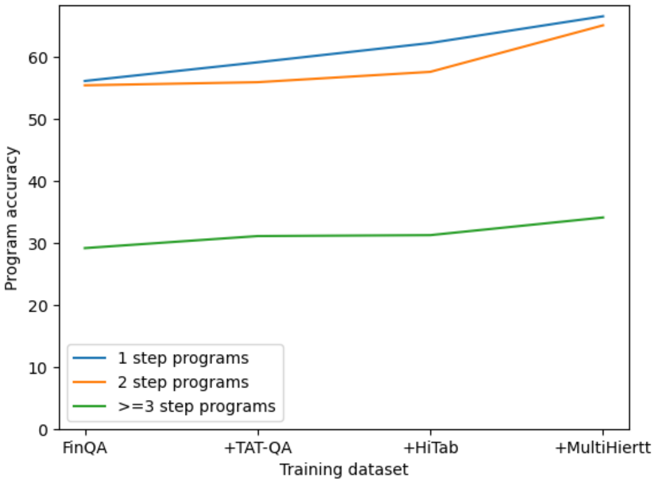


Figure 3.4: Program accuracy on the combined dataset as training datasets are added iteratively.

to relevant components of the input at each generation step. We demonstrated the effectiveness of our approach over four recently released tabular QA datasets. Our method, named CompAQT, was able to significantly improve program accuracy on three of the datasets, especially for multi-step programs. We also created a collection of QA samples for multi-step quantitative reasoning, by the datasets and unifying their format.

3.2 Counterfactual sampling for QA

Enterprise documents such as reports, forms, and analytical articles often include quantitative data in tabular form. The data in these tables can be self-contained, but more commonly the surrounding text provides more context that is necessary to understand the content. Answering questions over these hybrid tabular/text contexts requires reasoning that combines verbal and quantitative semantics.

Question answering over quantitative tabular/text data has gained recent traction with the release of datasets such as FinQA [26], TAT-QA [227], and HiTab [27]. Table 3.10 shows an example of a question that requires quantitative reasoning to derive the answer. Given the question and the tabular context, the output is a single-step program that leads to the final answer of `-20`.

A major challenge that state of the art models face is compositional generalization [119], especially when the number of reasoning steps grows [26]. In the context of quantitative QA, compositional generalization refers to the model's ability to generalize to new compositions of previously seen elements. As an example, if the model has encountered training examples that demonstrate calculations for "growth rate" and "percent change", we would like it to be able to come up with a reasonable hypothesis as to how to calculate "percent growth" or "rate of change". Table 3.9 demonstrates how this challenge becomes more difficult as the number of reasoning steps grows. For questions that require longer chains of reasoning, the model learns spurious patterns and unsuccessfully tries to leverage these memorized patterns to solve new

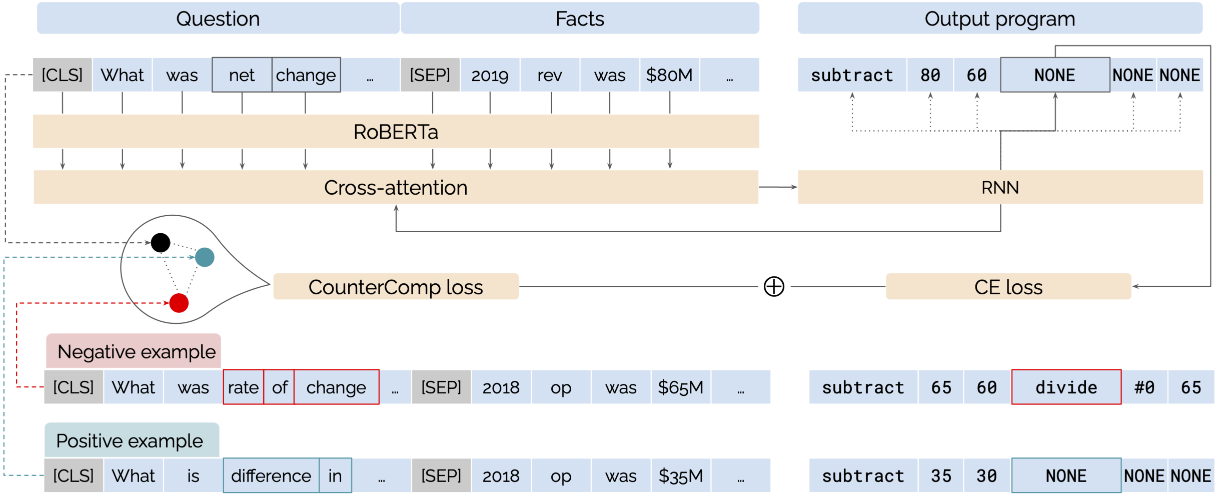

Question
Facts
Output program
[CLS] What was net change ... [SEP] 2019 rev was $80M ...
subtract 80 60 NONE NONE NONE
RoBERTa
Cross-attention
RNN
CounterComp loss
⊕
CE loss
Negative example
[CLS] What was rate of change ... [SEP] 2018 op was $65M ...
subtract 65 60 divide #0 65
Positive example
[CLS] What is difference in ... [SEP] 2018 op was $35M ...
subtract 35 30 NONE NONE NONE

Figure 3.5: A high-level illustration of our proposed method. The input example (anchor) is processed by cross-attention and recurrent modules to produce the output step by step. In addition to a regular Cross-Entropy loss, CounterComp adds an auxiliary triplet loss based on positive and negative examples. Note that the anchor and pos/neg examples are all processed through the RNN before calculating the triplet loss, a process which we have not illustrated due to space limitations. Also note that multiple pos/neg examples are sampled at each step.

problems.

# steps in output	% wrong operator(s)	% wrong operands	% wrong order of operands
1 step	39.07	53.64	7.28
2 steps	46.75	46.75	6.50
3 steps	56.47	29.41	14.12
>= 4 steps	52.00	40.00	8.00

Table 3.9: Share of FinQANet errors due to the selection of wrong operators or operands when applied to the FinQA dataset [26], broken down by the number of steps in the output.

The Table also shows that as the number of steps grows, generating the wrong operator becomes a more dominant mistake than selecting the wrong operand. Not only is this error more dominant, but it can also have a more destructive impact on the chain of reasoning, as it can derail the model's hidden representations from that point onward. As an example, our analysis of the FinQANet model [26] output showed that if the model generates an incorrect operator, it is about 30% more likely to commit other errors in the following steps compared to when the model generates an incorrect operand.

In this section, we propose CounterComp, an approach that can enhance compositional learning in multi-step quantitative QA. We take inspiration from the symbolic composition of arithmetic operations, and their correspondence to natural language phrases. Building on the work on attention alignments from previous studies, we propose an auxiliary metric learning loss that is focused on specific components of the input and output. Our sampling strategy is based on counterfactual scenarios. This means that the model learns proper representations for each component based on what-if scenarios. To the best of our knowledge, this is the first study that successfully applies component-wise counterfactual sampling as a metric learning strategy. We show how, when state of the art models are augmented with our auxiliary metric learning loss, they exhibit better performance in cases where multi-step reasoning is required. CounterComp outperforms current baselines on four recently released datasets, and show stronger performance on OOD samples.

<table>
<tr><td><b>Question</b></td><td>What was the net change in revenue from 2019 to 2020?</td></tr>
<tr><td><b>Tabular context</b></td><td><table>
<tr><th>Metric ($M)</th><th>2018</th><th>2019</th><th>2020</th></tr>
<tr><td>Operating expenses</td><td>35</td><td>29</td><td>30</td></tr>
<tr><td>Revenue</td><td>70</td><td>80</td><td>60</td></tr>
</table></td></tr>
<tr><td><b>Verbalized facts</b></td><td>2019 revenue was $80M.
2020 revenue was $60M.</td></tr>
<tr><td><b>Output program</b></td><td><code>subtract(80, 60)</code></td></tr>
<tr><td><b>Answer</b></td><td><code>-20</code></td></tr>
</table>

Table 3.10: Example of a quantitative QA problem over tabular data.

3.2.1 Background

The typical architecture of a quantitative QA model is composed of a retriever and a generator [69]. The retriever identifies the particular context where the answer might be found. Since the context can be a mix of table cells and sentences, often a tabular encoder [47] or verbalizer [26] is used to convert the cells into a natural language sequence. The retrieved context is referred to as retrieved *facts*. Next, the generator uses the question along with the facts to generate the output in a step by step fashion. In multi-step QA, the generator often combines a recurrent module with an attention mechanism [26], as illustrated in top half of Figure 3.5.

The output can be assessed in terms of program accuracy as well as execution accuracy. Our study is focused on improving program accuracy by encouraging compositional generalization in the generator.

There are two common approaches to improving compositional generalization. Attention alignment models encourage explicit alignments between natural language utterances (e.g. “rate of change”) and corresponding symbolic math operations (e.g. subtraction followed by division). Methods informed by counterfactuals use what-if scenarios to generalize to a wider variety of compositions and reduce the effect of memorization.

Attention alignments

Yin et al. [212] showed that additional supervision can be used to promote explicit alignments between components in the input and in the output. They added a regularization loss that encourages the cross-attention module to adjust its attention weights according to gold alignments. Using as few as 16 examples, their model was able to improve generalization in a semantic parsing task. CompAQT [123] extended this idea to multi-step quantitative QA. Instead of using additional supervision, it used natural language heuristics to create noisy alignment labels between input tokens and output symbols. The additional alignment loss improved the performance of three baseline models on multi-step reasoning tasks for four datasets.

Methods informed by counterfactuals

The success of alignment-based methods is limited by the fact that by heavily discouraging memorization, they underperform in settings where memorization can be helpful [125]. To strike a balance between memorization and generalization, one approach is to generate new training examples that cover important semantic gaps in the training data. This is reminiscent of how adversarial training can help better define the semantic contours of compositional representations [219]. Contrastive or metric learning methods pursue a similar goal, but instead of generating new samples, they leverage existing samples within the training set [61].

Counterfactual data augmentation (CAD) methods strive to achieve this by generating new samples using what-if scenarios [20, 105, 230]. This can be done by altering a minimally sufficient set of tokens in the input such that the output class changes [71]. There are two main challenges to creating these samples. First, it is difficult to identify the minimal set of tokens necessary to alter the output. Second, there is no guarantee that a counterfactual sample exists in the training set. To address these challenges, some studies employ human labelers [71] or a third party model [54]. In domains like semantic parsing and quantitative QA where the output is symbolic, an alternative approach leverages the structure of the output to avoid the need for human labelers. Li et al. [93] achieve this by intervening on the operands. Suppose that a question states "What was the net change in revenue from 2019 to 2020?" and the retriever produces two (verbalized) table cells: "2019 revenue was $80M" and "2020 revenue was $60M". The output program for this question would be: `subtract(80, 60)`. Given the numeric nature of the operands, it's possible to generate new scenarios such as "What if 2019 revenue was $90?" with the updated output `subtract(90, 60)`. Employing this method, Li et al. [93] augment the TAT-QA dataset [227] into a new dataset named TAT-HQA. They also enhance the verbal reasoning capacity of their model by offering the counterfactual scenario as a natural language prompt. Their model, named Learning to Imagine (L2I), outperforms state of the art models.

As mentioned in previously, models that struggle with compositional generalization suffer from errors in operator selection, whereas L2I is focused on the selection of operands. In this section, we present CounterComp, a method that focuses on counterfactual sampling for components that indicate operators[6]. Using natural language constraints from previous studies, we first find components that correspond to operators versus those that correspond to operands. Next, we use an auxiliary metric learning loss with positive and negative samples chosen based on those components. This helps us avoid the complexities associated with a data augmentation approach, such as the need for creation of additional human labels. The next section lays out our problem definition in more detail.

[6] Please refer to Appendix A.8 for a study on the use of CounterComp for operators versus operands.

3.2.2 Problem formulation

Let us consider the example provided by Table 3.11. Suppose Q is the question, represented as a sequence of tokens $q_1, \cdots, q_N$ (i.e. "what", "was", "the", $\cdots$, "2020", "?").

F is the evidence obtained by the retriever, made up of a sequence of tokens $f_1, \cdots, f_M$ (i.e. "2019", "revenue", "was", $\cdots$, "$60M").

The concatenation of these two sequences, i.e. $Q||F$, forms the input to the generator. The generator encodes $Q||F$ using a neural language model such as RoBERTa [107], resulting in an embedding matrix $\boldsymbol{U} \in \mathbb{R}^{d_{\text{enc}} \times (N+M)}$.

Consistent with Chen et al. [26], we represent the output S as a sequence of steps $s_1, \cdots, s_L$. Each step s_l can be an operator (such as `add` or `divide`), or an operand. Similar to Chen et al. [26], our programs are modeled as right-expanding binary trees with each operator having exactly two operands. If necessary, one or more operands are set to `NONE`, where `NONE` is a special constant. L is pre-defined as the maximum number of steps allowed. In the example from Table 3.11, S is: `subtract`, `80`, `60`, `NONE`, `NONE`, `NONE`.

To generate the lth output step s_l, the generator applies a cross-attention module to $\boldsymbol{U}$, resulting in the attention weight matrix $\boldsymbol{A}_l \in \mathbb{R}^{1 \times (N+M)}$ and the attention output $\boldsymbol{X}_l \in \mathbb{R}^K$. A recurrent module then generates the hidden vector $\mathbf{h}_l$, which is used to produce the output step s_l.

$$\begin{aligned} \mathbf{h}_l &= \text{RNN}(\mathbf{h}_{l-1}, \boldsymbol{X}_l) \\ \mathbf{s}_l &= \text{NN}(\mathbf{h}_l) \end{aligned} \tag{3.4}$$

where NN can be any neural module that projects $\mathbf{h}_l$ onto the simplex $\mathbf{s}_l \in \mathbb{R}^K$, from which s_l can be sampled: $s_l = \arg\max_k \mathbf{s}_{l,k}$. Our goal is to encourage $\mathbf{h}_l$ to be sensitive to the composition of the input $Q||F$ with regards to the current output step $\mathbf{s}_l$. This means that $\mathbf{h}_l$ needs to capture proper alignments between important terms in the input and the relevant operator/operand in the output.

To achieve this, we pursue a metric learning approach where positive and negative samples are generated according to counterfactual scenarios.

Counterfactual samples

Given a training example $([Q||F]^{(i)}, S^{(i)})$, we define an *intervention target* $\mathcal{Q}^{(i)}$ as a subsequence of the question tokens, i.e. $\mathcal{Q}^{(i)} = \{q_n^{(i)}; n \in \mathcal{N}^{(i)}\}$ where $\mathcal{N}^{(i)} \subseteq \{1, 2, \cdots, N\}$.

Suppose that changing the intervention target affects a single step in the output program $\mathcal{S}^{(i)} = s_l^{(i)}$, which we name the *intervention outcome*. Note that due to our focus on the generation of operators, we limit the intervention outcome to an operator. Since the output is composed of one operator followed by two operands followed by another operator and so on, l is selected from a limited index set: $l \in \{1, 4, 7, \cdots, L-3\}$. In the example from Table 3.11, the possible indices will be 1 and 4, representing the operators `subtract` and `NONE`.

Given this definition, it's possible to mine positive and negative examples for the ith training instance. A positive example $([Q||F]_{\text{pos}}^{(i)}, S_{\text{pos}}^{(i)})$ is an instance for which, despite a possible intervention in the target, the outcome remains the same, i.e. $\mathcal{Q}_{\text{pos}}^{(i)} \neq \mathcal{Q}^{(i)}$ and $\mathcal{S}_{\text{pos}}^{(i)} = \mathcal{S}^{(i)}$. A negative example $([Q||F]^{(i)}_{\text{neg}}, S^{(i)}_{\text{neg}})$ is an instance for which an intervention in the target leads to a change in the outcome, i.e. $\mathcal{Q}^{(i)}_{\text{neg}} \neq \mathcal{Q}^{(i)}$ and $\mathcal{S}^{(i)}_{\text{neg}} \neq \mathcal{S}^{(i)}$.

This allows us to define a triplet loss that encourages $\mathbf{h}^{(i)}_l$ to remain close to $\mathbf{h}^{(i)}_{l,\text{pos}}$ and far from $\mathbf{h}^{(i)}_{l,\text{neg}}$ with a margin of $\alpha^{(i)}$:

$$\mathcal{L}^{(i)}_{\text{triplet}} = \max\{||\mathbf{h}^{(i)}_l - \mathbf{h}^{(i)}_{l,\text{pos}}||^2_2 - ||\mathbf{h}^{(i)}_l - \mathbf{h}^{(i)}_{l,\text{neg}}||^2_2 + \alpha^{(i)}, 0\} \tag{3.5}$$

Figure 3.5 illustrates the sampling process for one training example. Note that this metric learning approach will only be valid if causal assumptions with regards to the intervention target are valid, i.e. the change in $\mathcal{S}^{(i)}_{\text{neg}}$ is in fact the result of the intervention in $\mathcal{Q}^{(i)}_{\text{neg}}$ and not a change in any other part of the input. In a data augmentation setting, this can be achieved by keeping the input fixed and perturbing a small segment that functions as the intervention target similar to [71]. However, as discussed in Section 3.2.1. this requires additional manual labor to annotate the perturbed examples.

In the next section, we describe how we impose certain constraints on the intervention target to achieve this in a self-supervised setting[7].

3.2.3 Methodology

Our goal is to identify potential positive and negative samples for the anchor $([Q||F]^{(i)}, S^{(i)})$. Suppose the anchor is the one shown in the top three rows of Table 3.11. Some terms are redundant between the question and the fact (i.e. "revenue", "2019", and "2020"). Those terms are often used by the retriever to find the correct facts. They are also used by the generator to find the correct order of operands.

There are also terms that are unique to the question, i.e. "What was the net change to", "from", and "to". In CompAQT, the authors showed that these can be used as indicators for the operators. Lastly, there are terms that are unique to the facts, i.e. "was $80M" and "was $60M". These can be used as indicators for the operands. We use these heuristics to guide our sampling strategy.

We flag all spans in the question that do not overlap with the facts, i.e. underlined blue segments. Those spans serve as candidate intervention spans. In the example from Table 3.11, this results in three candidates: "What was the net change in", "from", and "to".

Next, we seek a positive and a negative example within the training set. A positive example is a sample in which, despite possible changes in the question, the operators in the output remain consistent with the operators in the anchor. Table 3.11 shows one such example. Several terms have been altered in the question. However, we would only focus on the changes in the candidate spans. Here, "was" has changed to "is", "net change" has changed to "difference", "from" to "between" and "to" to "and". This results in a token-level Levenshtein distance of 5 (four edits and one insertion) [216]. We ignore the change from "revenue" to "operating expenses" and from "2019" to "2018", because those changes have occurred outside of our candidate spans and only correspond to operands.

[7] Note that the term "self-supervised" is used in this context to refer to the sampling strategy, i.e. no additional labeling is needed to generate the positive and negative samples.

A negative example is a sample in which exactly one output operator is altered, deleted, or added. Table 3.11 shows one such example. Here, the output includes a new operator `divide`. The question has also been altered with a token-level Levenshtein distance of 4.

The given positive and negative example can now be plugged into Equation 3.5. Instead of a fixed margin, we use the edit distances mentioned before to dynamically adjust the margin. Let $\text{NLD}^{(i)}_{\text{pos}}$ and $\text{NLD}^{(i)}_{\text{neg}}$ be the *normalized, token-level Levenshtein edit distance* between the anchor and the positive example, and the negative example, respectively. We set the margin to: $\alpha^{(i)} = 1 - |\text{NLD}^{(i)}_{\text{neg}} - \text{NLD}^{(i)}_{\text{pos}}|$

This encourages a larger margin for cases where the anchor is equally similar to the positive and the negative examples, and the model might have a harder time picking up on the nuances of each component.

Anchor	**Question**	What ***was*** the net change in ***revenue*** from ***2019*** to ***2020***?
	Facts	***2019 revenue was*** *$80M.* ***2020 revenue was*** *$60M.*
	Program	`subtract`(*80*, *60*) `NONE(NONE, NONE)`
	Candidate intervention spans	What the net change in from to
Positive sample	**Question**	What is the difference in ***operating*** ***expenses*** between ***2018*** and ***2020?***
	Facts	***2018 operating expenses*** *were $35M.* ***2020 operating expenses*** *were $30M.*
	Program	`subtract`(*35*, *30*) `NONE(NONE, NONE)`
Negative sample	**Question**	What ***was*** the rate of change of ***operating income from 2018*** to ***2019***?
	Facts	***2018 income from operating*** *activities* ***was*** *$65M.* ***2019 income from operating*** *activities* ***was*** *$60M.*
	Program	`subtract`(*65*, *60*) `divide`(`#0`, *65*)
Variables	$\mathcal{Q}^{(i)}$	$\{q^{(i)}_1, q^{(i)}_3, q^{(i)}_4, q^{(i)}_5, q^{(i)}_6, q^{(i)}_8, q^{(i)}_{10}\}$: what the net change in from to
	$\mathcal{S}^{(i)}$	$s^{(i)}_4$: `NONE`
	$\mathcal{Q}^{(i)}_{\text{pos}}$	$\{q^{(i)}_1, q^{(i)}_2, q^{(i)}_3, q^{(i)}_4, q^{(i)}_5, q^{(i)}_8, q^{(i)}_{10}\}$: what is the difference in between and
	$\mathcal{S}^{(i)}_{\text{pos}}$	$s^{(i)}_4$: `NONE`
	edit dist	5
	$\mathcal{Q}^{(i)}_{\text{neg}}$	$\{q^{(i)}_1, q^{(i)}_3, q^{(i)}_4, q^{(i)}_5, q^{(i)}_5, q^{(i)}_6, q^{(i)}_{12}\}$: what the rate of change of to
	$\mathcal{S}^{(i)}_{\text{neg}}$	$s^{(i)}_4$: `divide`
	edit dist	4

Table 3.11: Example of positive and negative sampling using counterfactual components. Blue underlined text indicates components that are unique to the question (candidates for intervention). These terms often indicate an operator. *Red italicized* text indicates terms that are unique to the facts. These terms often indicate operands. ***Bold italicized text*** indicates terms that are shared between the question and facts. These terms often indicate metrics.

Runtime optimization

There are two runtime challenges to this proposed approach: 1) Sampling can be costly if the entire training set has to be scanned for each batch. This means an online sampling strategy cannot be used. On the other hand, an offline strategy introduces a large overhead. A hybrid approach is needed. 2) Calculating the edit distance metric is a costly operation with $O(n^2)$ steps.

To solve the first problem, we build two indices prior to training. One index groups the samples by their sequence of output operators. This index can be used to sample positive examples.

The other index includes all training examples, and for each example, it includes the full list of one-step perturbations applied to its output operators. By generating all possible perturbations, we are able to find other samples whose outputs match the perturbed sequence (i.e. negative samples). For a sequence with n operators, all possible perturbations can be generated in $O(n \times K)$ time, where K is the number of possible operators[8].

Given the pre-generated positive and negative pools, we can also calculate and cache edit distances ahead of time. However, in practice, we realized that we could do so during training with little additional cost. This is because the edit distance is applied at the token-level[9], and is limited to candidate spans, rendering it relatively fast. The decision as to whether distances should be cached or calculated on the fly depends on the average size of each pool versus the number of training steps.

The algorithm outlined in Appendix A.7 summarizes our approach.

3.2.4 Experiments

Datasets

We use the hybrid CompAQT dataset, which is composed of four previously released datasets, namely **FinQA** [26], **TAT-QA** [227], **HiTab** [27], and **MULTIHIERTT** [224]. The authors filtered these four datasets down to QA pairs that require single or multi-step quantitative reasoning. They also processed the tables and outputs in all four datasets to match the FinQA format.

Baselines

We apply our proposed auxiliary loss to three baselines: 1) **FinQANet**, originally developed for the FinQA dataset [26]. 2) **TAGOP**, originally developed for the TAT-QA dataset [227]. 3) **Pointer-Verbalizer Network** (PVN), originally proposed by Nourbakhsh et al. [123]. We also apply the **CompAQT** loss to each model as a secondary baseline in order to determine how CounterComp compares to an attention-alignment strategy.

Sampling success rate

Another possible concern is that our sampling strategy might be limited, in that positive and negative samples might not always be available in the training set, or that limited availability of samples might bias the training process. To remediate the problem of unavailable samples, when a positive sample is missing, we use the anchor as the positive sample, and when a negative sample cannot be found, we use a uniformly sampled instance from the batch.

[8] Since we follow Chen et al. [26], in all of our experiments $K = 10$.

[9] Since we're using a language model that uses word-piece tokenization, in effect the runtime is at subword level.

Table 3.12 shows some statistics about the success rate of the sampling algorithm. "% Failure" identifies the share of training examples for which either a positive or a negative example was missing. Unsurprisingly this never happens for single-step programs, is very rare for two-step programs, and with the exception of HiTab, happens in less than 10% of the cases for longer programs. The Table also shows the average number of positive and negative examples found for each anchor. Again, HiTab has the lowest number of available samples, making it the most challenging dataset. In Section 3.2.5, we demonstrate how, even in cases with few possible samples, the model is able to generalize to unseen examples.

Dataset	**1 step**			**2 steps**			**3+ steps**		
	% Failure	**Avg. # pos samples**	**Avg. # neg samples**	**% Failure**	**Avg. # pos samples**	**Avg. # neg samples**	**% Failure**	**Avg. # pos samples**	**Avg. # neg samples**
FinQA	0	1457	3254	0.2	913	630	8.2	41	190
TAT-QA	0	1808	638	0	295	958	1.3	1055	66
HiTab	0	221	554	2.5	30	29	29.8	4	24
MULTIHIERTT	0	189	533	0.2	326	335	7.8	92	190

Table 3.12: The failure rate of sampling from each dataset (when no positive or no negative sample can be found for a given anchor), as well as the average number of positive and negative samples found for each anchor.

Settings

Since we are focused on the generator, in the experiments discussed in this section we will use gold facts and encode the input using RoBERTa-large [107][10]. We run the baselines with and without the additional $\mathcal{L}_{\text{triplet}}$ for 50 epochs with a learning rate of $5\text{e}{-5}$, the Adam optimizer [77] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. At each step, we sample (with replacement) 5 positive and negative pairs per anchor, and add the average auxiliary triplet loss to the main model loss with a weight of λ. After a grid search with a step-size of 0.1, we set λ to 0.4 for all experiments. All experiments were conducted on 8 NVIDIA T4 GPUS with 16 GBs of memory.

3.2.5 Results and analysis

Table 3.13 shows the program accuracy of baselines (top row of each cell) compared the addition of CounterComp loss (bottom row of eachcell). Among the baseline models, TAGOP is not designed to generate multi-step programs. Therefore we only apply it to the TAT-QA dataset, which has a set of pre-determined operations (e.g. `change ratio`). We also apply the PVN model to the combined dataset, but since FinQANet outperforms it on all benchmarks, we will continue to use FinQANet as the reference baseline model for the remaining experiments in this section.

As Table 3.13 shows, CounterComp consistently outperforms the baselines and the margin is often higher for longer programs. One notable exception is the TAT-QA dataset. As mentioned before, the dataset is not designed for open-ended multi-step reasoning and includes a limited set of possible operations. Therefore methods that encourage memorization might achieve higher

[10] Please refer to Appendix A.6 for results using retrieved facts.

performance on TAT-QA. HiTab is another challenging dataset, but despite low performance on longer sequences, CounterComp offers an improvement over the baseline.

Model	**Dataset**	**Program accuracy**			
		1 step	**2 steps**	**3+ steps**	**Overall**
TagOp	TAT-QA	45.01	39.56	42.73	43.25
+CompAQT		46.07	40.28	43.73	43.88
+CounterComp		**46.12**	**41.51**	**45.67***	**45.38**
PVN	Combined	68.14	61.33	13.54	56.64
+CompAQT		70.78	63.45	16.63	59.21
+CounterComp		**71.58***	**64.31***	**18.44***	**61.20***
FinQANet	FinQA	75.63	65.87	30.36	68.44
+CompAQT		78.68	75.12	35.85	73.74
+CounterComp		**79.13***	**75.45***	**36.86***	**74.49***
FinQANet	TAT-QA	**73.33**	63.76	64.88	**70.71**
+CompAQT		70.00	63.76	66.26	69.97
+CounterComp		70.56	**63.80**	**66.90**	70.01
FinQANet	HiTab	34.70	25.14	15.91	30.12
+CompAQT		34.73	29.94	17.35	32.23
+CounterComp		**34.94**	**30.00***	**17.39**	**32.61***
FinQANet	MultiHiertt	38.99	40.07	15.01	38.94
+CompAQT		38.87	42.35	16.77	40.82
+CounterComp		**39.25***	**42.51***	**16.86**	**40.85***
FinQANet	Combined	65.11	62.00	30.76	58.60
+CompAQT		66.60	65.14	34.16	60.88
+CounterComp		**67.91***	**66.00***	**36.91***	**61.82***
(Fixed margin)		66.19	64.89	34.06	59.58

Table 3.13: Program accuracy for program generation using baseline models v.s. using CompAQT loss, v.s. using CounterComp loss. * indicates that a gain/loss is significant at $p < 0.005$ compared to the baseline, using the paired-bootstrap test proposed by Berg-Kirkpatrick et al. [12] for $b = 10^3$.

Auxiliary triplet loss versus auxiliary attention alignment loss

The middle row of each cell in Table 3.13 shows the program accuracy when CompAQT loss is added instead of CounterComp loss. As previously described, CompAQT imposes an auxiliary attention alignment loss such that tokens related to operators receive more attention during the generation of operators. Even though this leads to improvements over the baselines, CounterComp outperforms CompAQT in all experiments. This might be due to the fact that the regularizing effect of CompAQT loss is not as strong as the representation learning impact of CounterComp.

Despite the fact that CounterComp was not designed as an attention alignment model, it does have an impact on how attention patterns evolve during training. Table 3.14 shows the top-attended input tokens during the generation of a `divide` operator in various contexts. For a singular division operation, FinQANet attends to tokens such as “year” whereas CounterComp encourages the model to attend to more relevant tokens such as “net” and “change”. A subtraction followed by a division often indicates a percentage calculation, as captured by both models. An addition followed by a division often indicates an average calculation. Again, CounterComp is able to capture relevant tokens but the FinQANet baseline seems to attend to some memorized tokens such as “annual”.

Model	**Top attended tokens during the generation of `divide`**		
	`divide`	`subtract` **`divide`**	`add` **`divide`**
FinQANet	share, year	ratio, percent	annual, per
+CounterComp	net, change	share, percent	average, per

Table 3.14: Top attended tokens during the generation of the division operator in various sequences. The dataset used for this experiment is FinQA.

Fixed versus adaptive margin

The last row of Table 3.13 shows the performance of the FinQANet model on the combined dataset using the CounterComp loss with a fixed margin of 1. The performance suffers, especially as the number of steps grows. This further demonstrates the importance of the adaptive margin $\alpha^{(i)}$ that takes the edit distance into account.

Question	Evidence	Gold program	FinQANet	FinQANet + CounterComp

What was the gross margin decline in fiscal 2004 from 2003?	1) the gross margin pct of 2004 is 27.3%
	2) the gross margin pct of 2003 is 27.5%
		`subtract(27.5, 27.3)`	`subtract(27.5, 27.3),`
			`divide(#0, 27.5)`
				`subtract(27.5, 27.3)`

What percentage of amounts expensed in 2009 came from discretionary company contributions?	1) amounts expensed for 2009 was $35.1
	2) expense includes a discretionary company contribution of $3.8
		`divide(3.8, 35.1),`
		`multiply(#0, const_100)`
			`divide(3.8, 35.1)`	`divide(3.8, 35.1),`
				`multiply(#0, const_100)`

Did the share of securities rated aaa/aaa increase between 2008 and 2009?	1) the aaa/aaa share of 2009 is 14%
	2) the aaa/aaa share of 2008 is 19%
		`greater(14, 19)`	`subtract(14, 19),`
			`subtract(14, #0)`
				`greater(14, 19)`

On February 13, 2009 what was the market capitalization?	1) on February 13, 2009, the closing price of our common stock was $28.85 per share 2) as of February 13, 2009, we had 397097 outstanding shares of common stock	`divide(397097, 28.85)`	`multiply(397097, 28.85)`	`multiply(397097, 28.85)`

Table 3.15: Four examples from the FinQA dataset, showing CounterComp's success and failure in capturing compositional expressions. Note that some numbers have been truncated to save space.

Model	**Program accuracy on test dataset**			
	TAT-QA	**HiTab**	**MultiHiertt**	**FinQA (unseen programs)**
FinQANet	41.64	22.80	35.33	65.74
+CompAQT	39.88	22.71	35.28	70.32
+CounterComp	**42.00**	**22.97**	**36.94**	**73.53**

Table 3.16: OOD performance of FinQANet variations when trained on the FinQA dataset and tested on other datasets, or tested on unseen operator compositions in the FinQA dev set.

Qualitative examples

Table 3.15 shows four qualitative examples from the FinQA dataset. The first two rows show how CounterComp enables the FinQANet model to represent concepts such as "decline" and "percentage" more accurately. The third example shows how CounterComp is able to determine the difference between a calculation question and a yes/no question. The last row shows a failure example, where CounterComp does not improve the performance of FinQANet. This particular example requires deep domain expertise to address. This highlights the need for methods that allow domain expertise to be represented more effectively [26].

Compositional v.s. OOD generalization

In a recent study Joshi and He [68] showed that current approaches to counterfactual data augmentation do not necessarily lead to better generalization to out-of-distribution (OOD) samples. To test whether this holds for CounterComp, we conduct two studies. First, we train FinQNet with and without CounterComp loss on the FinQA dataset, then test it on the other three datasets. Note that the four datasets are based on different domains. FinQA and TAT-QA are both based on financial reports, but while FinQA was derived from US filings, TAT-QA is based on international filings and therefore covers a wider variety of metrics. HiTab and MULTIHIERTT are both based on other types of corporate reports with highly complex tabular structures.

The first three columns of Table 3.16 show a slight improvement when CounterComp is used in this setting. In contrast, using CompAQT loss slightly hurts the performance, demonstrating CounterComp's higher OOD generalization potential.

Next, we select a subset of samples from the FinQA dev set that have unseen compositions compared to the training set. This means that the particular combination of operations were never seen during training. As the last column of the table shows, CounterComp outperforms the baseline by more than 7 points. This further demonstrates how improving representation learning at the component level can enhance generalization to unseen contexts.

3.2.6 Conclusion

In this section, we presented CounterComp, a method that leverages counterfactual contrast to enable metric learning for quantitative QA. We show how using the auxiliary CounterComp loss can improve compositional generalization in multi-step reasoning tasks, especially as the number of steps grows.

Due to runtime challenges, we proposed a hybrid offline/online sampling strategy that uses pre-defined indices for easier lookup operations. This allows us to capture samples that have a contrast of one operator with the anchor. In future studies, we hope to capture contrastive samples with longer perturbation chains. We also hope to examine the effectiveness of counterfactual compositional contrast in other domains such as semantic parsing and question answering over multimodal input.

Lastly, we hope to extend the use of CounterComp to enhance the performance of the retriever, using the heuristics introduced in Section 3.2.3 (i.e. by focusing on components in the question that overlap with the facts). This can result in a quantitative QA pipeline that is powered by compositional contrast in an end-to-end fashion.

Chapter 4

Grounded structures for multimodal fusion

Let us return to the scenario that we introduced in Section 1.1 about Alice, a knowledge worker at a financial firm who is tasked with processing authorized signatory forms. As we discussed in that Section, Alice would like to automate the task of extracting key information and relations from the forms in a grounded fashion. This would require a model that:

1. Can detect key relations as well as entities.
2. Produces grounded outputs, i.e. outputs that can be located within the input using bounding boxes or other references.

In this chapter, we introduce two new methodologies that tackle the above challenges by encouraging grounded spatio-visual reasoning. Table 4.1 places our contributions in the space of grounding methodologies introduced in Section 2.4.

Reasoning	**Grounding the design**	**Grounding the objective**	**Grounding the search space**
Quantitative	Graph-based representation [148, 220]	Attention coverage [125] Supervised alignments [212] Unsupervised alignments (Section 3.1) Metric learning (Section 3.2)	Data augmentation via counterfactuals [94] Counterfactual sampling (Section 3.2)
Spatio-visual	Spatial-aware attention [205] Graph-based representation [31, 87, 88] Graph-based generation (Section 4.2) Modality-adaptive attention [95]	Cell recovery [47, 211] Masked column prediction [211] MVLM [9, 207] TIA/TIM [9, 56, 205] LTR [9, 131] Cluster membership (Section 4.1)	Probabilistic soft logic [169] Data augmentation via perturbations [136]

Table 4.1: An updated view of Table 2.3, where our proposed methods have been added to corresponding cells, blue. Each highlight has a reference to the section where it is covered.

4.1 Graph-inspired representations for visual information extraction

Layout-aware language models have been used to create multimodal representations for documents that are in image form, achieving relatively high accuracy in document understanding tasks. However, the large number of parameters in the resulting models makes building and using them prohibitive without access to high-performing processing units with large memory capacity. We propose an alternative approach named APReCoT that can create efficient representations without the need for a neural visual backbone. This leads to an 80% reduction in the number of parameters compared to the smallest SOTA model, widely expanding applicability. In addition, our layout embeddings are pretrained on spatial and visual cues alone, and only fused with text embeddings in downstream tasks, which can facilitate applicability to low-resource or multi-lingual domains. Despite using 2.5% of training data, we show competitive performance on two form understanding tasks: semantic labeling and link prediction.

APReCoT, which stands for Alignment, Proximity, REpetition, and COntrast-aware Transformer, is inspired by the principles of layout design that were introduced in Section 1.2.2. By carefully designing multimodal features that represent these principles, APReCoT is able to perform spatio-visual reasoning with fewer parameters and smaller training data.

4.1.1 Background

Layout-aware language models represent documents as multimodal artifacts, composed of visual and textual content. By jointly modeling the stylistic, spatial, and semantic cues, they capture constructs such as hierarchy, correspondence, enumeration, and tabulation. In this section, we present an approach that combines the expressivity of large multimodal networks with the efficiency of graph networks. This addresses three major challenges that transformer-based models face:

1. The use of large neural backbones for extracting visual features vastly increases the size and memory requirements for such models. For LayoutLMv2 for example, a NVIDIA GTX 1080 GPU with 12 GBs of memory fails to accommodate a batch size of 8. On single-GPU platforms, this means that fine-tuning the model alone can take prohibitively long.
2. The visual feature extractor does not free the model of reliance on OCR engines for extracting text and bounding boxes. This means that the preprocessing step remains slow and memory intensive.
3. The transformer architecture often uses a variation of Masked Language Modeling (MLM) [35], which does not make efficient use of layout cues, requiring the model to be pretrained on very large corpora. A common pretraining dataset is IIT-CDIP [44], composed of several million documents. In low-capacity environments, preprocessing a corpus at this scale can take weeks, further prohibiting the wider research community from adopting and experimenting with the models.

In contrast, our proposed method achieves competitive performance with a fraction of parameters, memory requirements, and pretraining data. Instead of a neural visual backbone, we generate informative layout features using the OCR output alone. APReCoT takes inspiration from recent graph-based approaches, but instead of an explicit graph representation with a highly customized design, we extend the spatial-aware attention mechanism [205] to capture contrast in spacing as well as style. Furthermore, we replace the MLM objective with a metric learning paradigm that captures concepts such as proximity, alignment, and correspondence. This leads to competitive performance in two form understanding tasks, namely semantic labeling as well

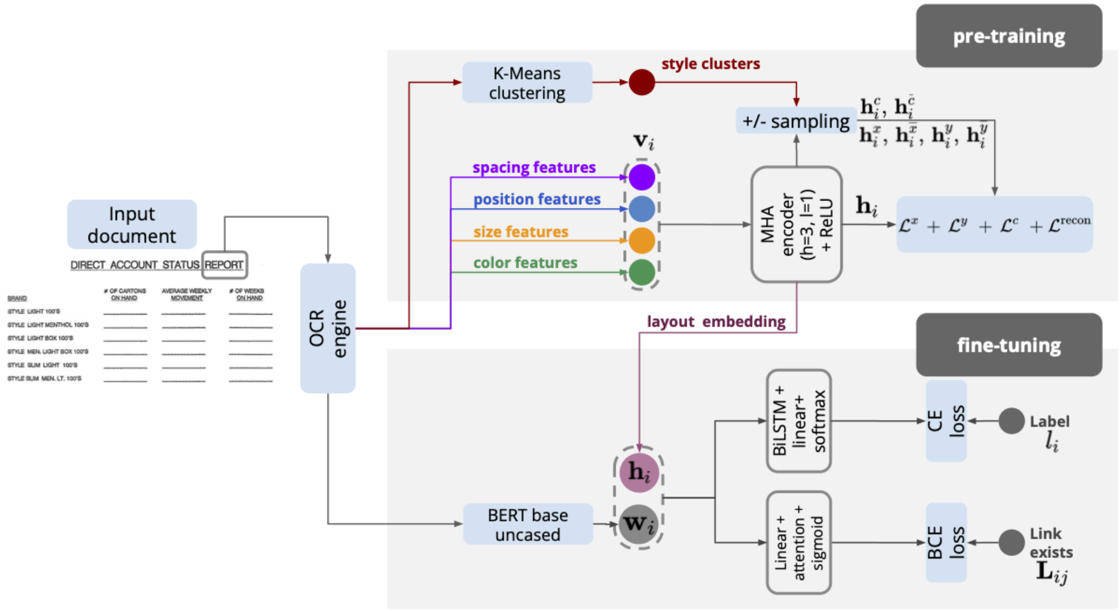

pre-training
K-Means clustering
style clusters
+/- sampling
$\mathbf{h}_i^c$, $\mathbf{h}_i^{\bar{c}}$
$\mathbf{h}_i^x$, $\mathbf{h}_i^{\bar{x}}$, $\mathbf{h}_i^y$, $\mathbf{h}_i^{\bar{y}}$
$\mathbf{v}_i$
spacing features
position features
size features
color features
MHA encoder (h=3, l=1) + ReLU
$\mathbf{h}_i$
$\mathcal{L}^x + \mathcal{L}^y + \mathcal{L}^c + \mathcal{L}^{\text{recon}}$
Input document
DIRECT ACCOUNT STATUS REPORT
OCR engine
layout embedding
fine-tuning
BiLSTM + linear+ softmax
CE loss
Label l_i
$\mathbf{h}_i$
$\mathbf{w}_i$
BERT base uncased
Linear + attention + sigmoid
BCE loss
Link exists $\mathbf{L}_{ij}$

Figure 4.1: APReCoT's architecture during pre-training and fine-tuning.

.

as link prediction.

4.1.2 Methodology

Figure 4.1 illustrates the end-to-end architecture of APReCoT. As the figure shows, we follow a three-step process. First, input documents are processed using an OCR engine, and relevant layout features are extracted. Second, the layout features are transformed into layout-aware embeddings through a self-supervised pretraining step. Finally, the layout-aware embeddings are combined with text embeddings and fine-tuned on downstream tasks. The remainder of this section describes each step in detail.

Preprocessing and token representation

We process each document through an OCR engine, which converts each page into a sequence of tokens $t_1, t_2, ..., t_N$ and corresponding bounding boxes $b_1, b_2, ..., b_N$. To represent a token t_i (with corresponding bounding box b_i), we construct a vector by concatenating the following features:

- **Position features**: The top and left coordinates of b_i, as well as the x and y coordinates of its centroid.
- **Spacing features**: The spacing between b_i and its immediate neighbors to the left, right, top, and bottom, as well as the number of neighbors in b_i's line-of-sight to the left, right, top, and bottom[1].
- **Size features**: The height and width of b_i as well as the average character width in b_i.

[1] Line-of-sight is defined according to the formulation provided by [30].

- **Color features**: The RGB values of the foreground and background colors of b_i, obtained by applying K-Means clustering (K=2) to the distribution of RGB values within each bounding box[2].

All features are concatenated into a vector $\mathbf{v}_i \in \mathbb{R}^{21}$, and normalized over minimum and maximum values on the page[3]. This allows us to order the $\mathbf{v}_i$s by their position using a simple algorithm: 1) To approximate horizontal lines, we group $\mathbf{v}_i$s whose bottom coordinates are within 1 percentile of each other. 2) We order the $\mathbf{v}_i$s first by line number, then by left coordinates. This results in a sequence of vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N$ representing each page, where $\mathbf{v}_1$ corresponds the top-left token and $\mathbf{v}_N$ corresponds to the bottom-right token [4].

Layout-aware pretraining

To create rich representations of layout information, we refer back to the design principles of contrast, proximity, alignment, and repetition, introduced in Section 1.2.2. The below subsections describe how we capture each principle in APReCoT.

Capturing proximity and contrast As previously mentioned, contrast can manifest itself in position, size, style, or color. The literature on self-attention has so far mainly focused on capturing positional contrast (i.e. proximity). [35] use additive 1-D positional encodings [184] for text sequences. Since layouts carry two dimensional information, some studies have extended this idea to include 2-D encodings [207]. However, additive 2-D encodings do not capture relative positions very well [161]. A simple yet effective alternative was proposed by [144], who use an additional bias term to capture relative positions. [205] extend this idea to create a spatial-aware attention mechanism, where scaled dot-product attention between queries and keys is augmented by three bias terms– representing relative 1-D position, relative x-axis position, and relative y-axis position.

We further extend this idea to capture relative information along multiple dimensions, namely position, spacing, size, and color. This allows the model to focus on contrast along all of these dimensions instead of position alone. We refer to this approach as Layout-Aware Attention. Concretely, we modify scaled dot-product attention to follow the below equation:

$$\alpha_{ij} = \frac{1}{\sqrt{d_{head}}}(\mathbf{v}_i\mathbf{W}^Q)(\mathbf{v}_j\mathbf{W}^K)^\intercal \tag{4.1}$$

where d_{head} is the number of attention heads, $\mathbf{W}^Q$ is the query weight matrix, $\mathbf{W}^K$ is the key weight matrix, and α_{ij} is the attention weight for $\mathbf{v}_i$ and $\mathbf{v}_j$. We calculate the augmented attention weight α'_{ij} as:

$$\alpha'_{ij} = \alpha_{ij} + \sum_f \mathbf{b}_{f_j - f_i} \tag{4.2}$$

[2] Note that CDIP and FUNSD datasets are composed of grayscale documents, but NAF includes color. However even for grayscale documents, color distribution can help distinguish different styles.

[3] Feature values are normalized over each page because all of the datasets used in this study are limited to single-page examples. The same methodology can be applied at document level.

[4] The left-to-right-and-top-to-bottom order can be redefined depending on the language of interest. Note that this method need not be robust against all compositions (e.g. multi-column pages) and is only applied to provide rough ordering information.

where $\mathbf{b} \in \mathbb{R}^M$ is a bias vector, and $\mathbf{b}_{f_j - f_i}$ represents the difference in the values of the fth feature of $\mathbf{v}_i$ and $\mathbf{v}_j$, binned into M groups. Finally, the output is calculated as:

$$\mathbf{h}_i = \sum_j \frac{\exp(\alpha'_{ij})}{\sum_k \exp(\alpha'_{ik})} \mathbf{v}_j \mathbf{W}^V \tag{4.3}$$

where $\mathbf{W}^V$ is the value weight matrix.

Capturing alignment Horizontal or vertical alignment plays an important role in determining semantic relationships between text segments. To capture this, we encourage the representations learned by the encoder to reflect these alignments, using a triplet loss function. For each hidden representation $\mathbf{h}_i$, we sample from the set of tokens that align with it horizontally ($\mathbf{h}_i^x \in \{\mathbf{h}_j; \mathbf{v}_j^{(x)} = \mathbf{v}_i^{(x)}\}$), and the tokens that do not align with it horizontally ($\mathbf{h}_i^{\overline{x}} \in \{\mathbf{h}_j; \mathbf{v}_j^{(x)} \neq \mathbf{v}_i^{(x)}\}$). We then use a triplet margin loss function to encourage $\mathbf{h}_i$ to be close to $\mathbf{h}_i^x$ and far from $\mathbf{h}_i^{\overline{x}}$:

$$\mathcal{L}^x = \tfrac{1}{N} \textstyle\sum_{i=1}^N \max\{||\mathbf{h}_i - \mathbf{h}_i^x|| - ||\mathbf{h}_i - \mathbf{h}_i^{\overline{x}}|| + 1, 0\}$$

Similarly, we calculate a triplet loss along the vertical dimension:

$$\mathcal{L}^y = \tfrac{1}{N} \textstyle\sum_{i=1}^N \max\{||\mathbf{h}_i - \mathbf{h}_i^y|| - ||\mathbf{h}_i - \mathbf{h}_i^{\overline{y}}|| + 1, 0\}$$

In practice, instead of enforcing exact equality, we allow the horizontal and vertical alignment of positive samples to be different from the anchor's within a margin of $[-0.01, 0.01]$. For all other features, we allow a larger margin of noise, i.e. $[-0.1, 0.1]$. This method, inspired by [145], allows us to generate positive and negative samples synthetically, alleviating the computational bottleneck that sampling poses.

Capturing repetition As previously mentioned, document layout often follows a consistent pattern of styles. This limits the number of shapes, sizes and colors used in each document. Tokens that have similar style often play a similar function in the document. To capture this, we apply K-Means clustering to the style and color features of each sample to obtain K clusters. Each $\mathbf{h}_i$ will thus be associated with a cluster c_i. We then sample other tokens that share the same cluster and those that don't, leading to a third loss:

$$\mathcal{L}^c = \tfrac{1}{N} \textstyle\sum_{i=1}^N \max\{||\mathbf{h}_i - \mathbf{h}_i^c|| - ||\mathbf{h}_i - \mathbf{h}_i^{\overline{c}}|| + 1, 0\}$$

The total loss is then formulated as:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^x + \mathcal{L}^y + \mathcal{L}^c + \mathcal{L}^{\text{recon}} \tag{4.4}$$

where $\mathcal{L}^{\text{recon}}$ is the reconstruction loss, calculated as the Mean Squared Error between $\mathbf{v}_i$ and the final output of the attention layer FFN(AddNorm($\mathbf{h}_i$)). This term functions as a regularizer, encouraging the layout embeddings to not stray too far from the original feature space.

Fusion and fine-tuning

The outcomes of the pretraining process are layout-aware embeddings. To add text, we simply concatenate the layout and text embeddings together, i.e. $\mathbf{e}_i = \mathbf{h}_i \parallel \mathbf{w}_i$, where $\mathbf{h}_i$ is the layout embedding for token t_i, $\mathbf{w}_i$ is its text embedding obtained using a pretrained language model, and $\parallel$ represents concatenation. The resulting $\mathbf{e}_i$s can be used for spatial reasoning tasks over documents. In this study, we explore two such tasks–semantic labeling and link prediction in forms.

Semantic labeling This task involves token-level classification over four categories: "header", "question" (a.k.a field name), "answer" (a.k.a field value), and "other". We use a minimal architecture–a single-layer Bidirectional LSTM module followed by a linear projection. We then use cross entropy loss to train the model for token classification:

$$\mathcal{L}^{\text{sl}} = \frac{1}{N} \sum_{i=1}^{N} \text{CE}(\text{Softmax}(\text{FFN}(\text{BiLSTM}(\mathbf{e}_i))), l_i) \tag{4.5}$$

where l_i is the label corresponding to $\mathbf{e}_i$.

Link prediction Links within a form convey correspondence, e.g. field values are linked to their corresponding field names, and field names to their corresponding headers. The main challenge of the link prediction task lies in extreme class imbalance, i.e. only a few links exist within the quadratic space of possible links between any given pair of tokens. We choose to model this quadratic space using the self-attention mechanism.

Concretely, we use a linear projection to reduce the size of $\mathbf{e}_i$ vectors to a smaller dimension P. Next, we use a single self-attention layer as an activation function. This layer produces two outputs—attention output $\mathbf{O}^A \in \mathbb{R}^{N \times P}$ and attention weights $\mathbf{W}^A \in \mathbb{R}^{N \times N}$ where N is the document length. We apply binary cross entropy loss on $\mathbf{W}^A$ to determine whether a link should exist between any given pair of tokens:

$$\mathcal{L}^{lp} = \frac{1}{N^2/2} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{BCE}(\sigma(\mathbf{W}^A_{ij}), \mathbf{L}_{ij}) \tag{4.6}$$

where $\mathbf{L}_{ij}$ is a binary matrix identifying whether tokens t_i and t_j should have a link. Figure 4.1 illustrates the end to end pipeline. In both tasks, the fine-tuned model has fewer than 3 million parameters. This includes all parameters in the layout-aware pretraining model, plus the parameters in the task-specific model for semantic labeling or link prediction.

4.1.3 Experiments and results

To demonstrate the effectiveness of our approach, we pretrain APReCoT on a collection of low-quality image documents, and apply it to two downstream tasks, namely semantic labeling and link prediction. As previously mentioned, RVL-CDIP [44] is a commonly used dataset for pretraining layout-aware models. It is composed of 1 million image documents with diverse layouts and visual quality. To demonstrate the data-efficiency of our approach, we choose a small subset for pretraining, namely the subset of documents tagged as forms. This amounts to 25,003 samples or about 2.5% of the complete dataset[5].

To obtain tokens and bounding boxes, we process the documents using the Tesseract OCR engine [2]. We use one multi-headed self attention layer with 3 heads and a hidden size of 120. We set the number of bins for relative biases to $M = 11$, the number of style clusters to $K = 12$ and maximum sequence length to $N = 512$ tokens. To represent text, we use 768-dimensional BERT embeddings [35]. We pretrain APReCoT with a batch size of 64 for 5 epochs, using the AdamW optimizer [76] with a learning rate of $1\mathrm{e}{-2}$, gradient clipping of $5\mathrm{e}{-1}$, weight decay of $1\mathrm{e}{-2}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. During fine-tuning, without any change to the parameters, we continue to train the model on the training split of the dataset of interest for an additional 5 epochs before freezing the embeddings[6]. On one NVIDIA 1080 GPU, the pretraining process takes roughly two hours.

Datasets

We use two datasets that represent form understanding tasks, namely FUNSD [64] and NAF [30]. FUNSD is composed of 200 noisy scanned forms, split between 150 training and 50 test examples. NAF includes scanned forms with complicated layouts and overlapping or handwritten text segments. The dataset offers 682 training, 59 validation, and 63 testing examples. Both datasets support annotations for token-level semantic labeling as well as link prediction tasks. FUNSD provides 4 labels for each token, namely "header", "question", "answer", and "other". NAF includes 14 classes, with a more granular breakdown of categories of field names and field values such as checkboxes, number fields, text fields, etc. Both datasets also provide link annotations that show correspondence between answers (field values) and questions (field names) as well as correspondence between questions (field names) and headers.

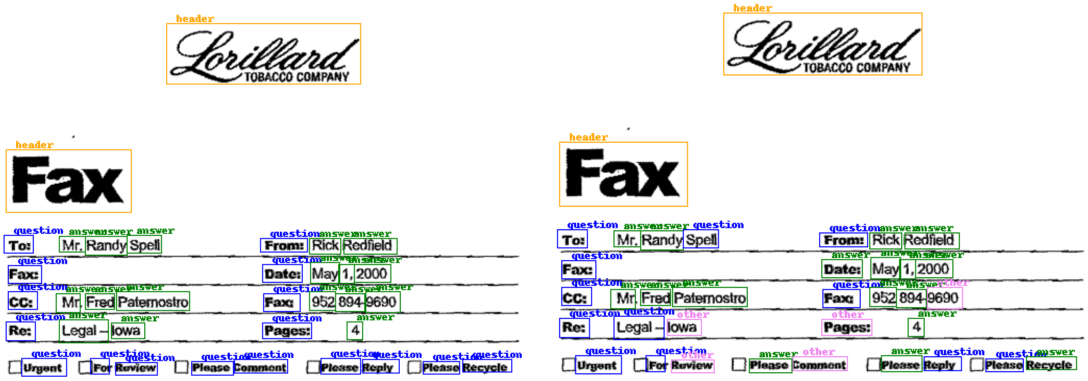
Semantic labeling

We train the BiLSTM-based module for 100 epochs with the same optimizer as pretraining, a learning rate of $5\mathrm{e}{-3}$ and gradient clipping of $5\mathrm{e}{-1}$. Table 4.2 shows the F1 performance on the FUNSD dataset. Despite an 80% reduction in the number of parameters, APReCoT's performance is comparable to Word-FUDGE [31], an extension of Visual FUDGE that uses native bounding boxes. To the best of our knowledge, there is no current benchmark on semantic labeling on the NAF dataset. We report a F1 score of 59.07 (P=56.80, R=61.54) on the 14-class labeling task.

Figure 4.2 shows two qualitative examples of semantic labeling results from the FUNSD dataset. As the Figure illustrates, APReCoT is more effective at acknowledging horizontal alignment than vertical alignment. In the top example the word "Date" has been labeled as an answer, possibly due to its uninterrupted alignment with "Fax". APReCoT also shows vulnerability to sequential dependencies that range beyond 2-3 tokens. This indicates that a stronger proximity

[5] Since FUNSD is a subset of RVL-CDIP, we remove the 50 test examples from the CDIP dataset.

[6] Note that FUNSD and NAF are much smaller than RVL-CDIP and do not introduce large instabilities during continuous learning.



(a) True labels.

(b) Model predictions.

Figure 4.2: Semantic labeling example from the FUNSD dataset.Each category has been color-coded. Yellow: header, blue: question, green: answer, pink: other.

bias such as a segment detection loss introduced during pretraining might improve APReCoT's results.

Model	**P**	**R**	**F1**	**# Params**
BERT$_{\text{BASE}}$**	54.69	67.10	60.26	110M
LayoutLM$_{\text{BASE}}$*	75.97	81.55	78.66	113M
BROS*	**80.56**	81.88	81.21	138M
LayoutLMv2$_{\text{BASE}}$*	80.29	**85.39**	**82.76**	200M
Word-FUDGE*	69.37	75.30	72.21	17M
APReCoT	70.42	71.13	70.77	3M

* Results reported in [31].

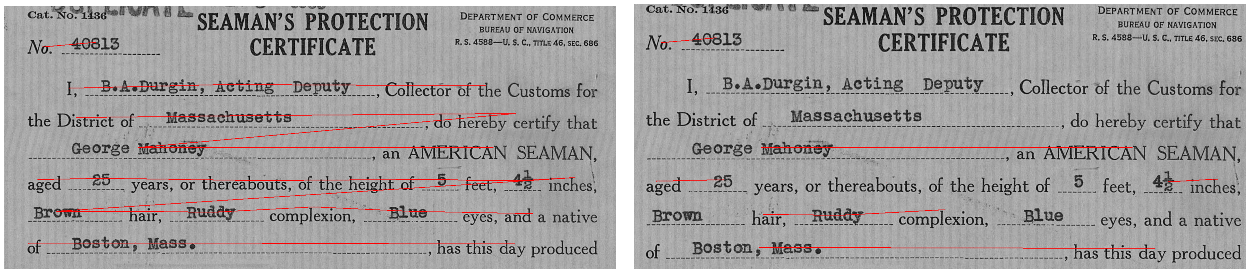
** Results reported in [205].

Table 4.2: Semantic labeling results for the FUNSD dataset.

Link prediction

As described in section 4.1.2, we use self-attention to train the model to predict whether a link exists between a given pair of tokens. We train the model for 100 epochs with a learning rate of $5e{-}3$. Table 4.3 shows the F1 performance on the FUNSD dataset. APReCoT is able to outperform LayoutLM$_{\text{BASE}}$, which has 35 times as many parameters and is trained on the IIT-CDIP dataset [92] with more than 6 million documents. It is outperformed by BROS (with 138M parameters) and Word-FUDGE, a graph-based method designed for link prediction.

Table 4.4 shows the results on the NAF dataset, which offers a simple version (only links between field names and values), and full version (all links). APReCoT has lower performance on NAF compared to FUNSD. This could be due to the data distribution in FUNSD being closer to the pretraining dataset (RVL-CDIP). Both include grey-scale forms with the majority of bounding boxes in horizontal or vertical alignment. Whereas NAF includes a wider color diversity, stamped and hand-written content, and polygons with various orientations.



Cat. No. 1136 SEAMAN'S PROTECTION CERTIFICATE DEPARTMENT OF COMMERCE BUREAU OF NAVIGATION R. S. 4588—U. S. C., TITLE 46, SEC. 686

No. 40813

I, B.A.Durgin, Acting Deputy, Collector of the Customs for the District of Massachusetts, do hereby certify that George Mahoney, an AMERICAN SEAMAN, aged 25 years, or thereabouts, of the height of 5 feet, 4½ inches, Brown hair, Ruddy complexion, Blue eyes, and a native of Boston, Mass., has this day produced

(a) True links.

Cat. No. 1136 SEAMAN'S PROTECTION CERTIFICATE DEPARTMENT OF COMMERCE BUREAU OF NAVIGATION R. S. 4588—U. S. C., TITLE 46, SEC. 686

No. 40813

I, B.A.Durgin, Acting Deputy, Collector of the Customs for the District of Massachusetts, do hereby certify that George Mahoney, an AMERICAN SEAMAN, aged 25 years, or thereabouts, of the height of 5 feet, 4½ inches, Brown hair, Ruddy complexion, Blue eyes, and a native of Boston, Mass., has this day produced

(b) Model prediction.

Figure 4.3: Link prediction example from the NAF dataset.

Figure 4.3 includes a link prediction example from the NAF dataset. As the figure demonstrates, APReCoT often fails to recover links across multiple lines. It also shows a preference for short-distance links, possibly due to this pattern being more common in the pretraining dataset. A possible way to address these issues is to use a different sampling technique during metric learning, to allow for larger variance for negative samples and select longer-range peers as positive samples.

Model	**P**	**R**	**F1**	**# Params**
LayoutLM$_{\text{BASE}}$*	41.29	44.45	42.81	113M
BROS*	**64.30**	**69.86**	**66.96**	138M
Word-FUDGE*	58.08	67.83	62.58	17M
Ours	52.07	57.32	54.57	3M

* Results reported in [31].

Table 4.3: Link prediction results for the FUNSD dataset.

Model	**Version**	**P**	**R**	**F1**	**# Params**
Visual FUDGE*	Simple	**63.60**	**73.20**	**68.05**	17M
Visual FUDGE*	Full	**59.92**	**54.92**	**57.31**	17M
APReCoT	Simple	51.15	52.10	51.62	3M
APReCoT	Full	46.26	44.67	45.45	3M

* Results reported in [31].

Table 4.4: Link prediction results for the NAF dataset.

4.1.4 Ablation studies

To determine the impact of each component in the system, we perform the following ablation studies on the FUNSD dataset. The results are listed in order in Table 4.5:

1. Using no layout-aware embeddings, we use 768-dimensional BERT$_{\text{BASE}}$ embeddings with the same architectures that were described in Section 4.1.2—BiLSTM for semantic labeling and attention activation for link prediction.

2. We replace the pre-trained layout token $\mathbf{h}_i$ with the raw feature vector $\mathbf{v}_i$ without any pretraining.
3. We replace the pretraining process with a modified version of MVLM, which we refer to as Masked Layout Language Modeling (MLLM). In each sample, 20% of $\mathbf{v}_i$s are replaced with random feature vectors and the encoder is trained to recover them using MSE loss between the attention output and the original feature vectors[7].
4. We use the full model but disable the alignment loss terms $\mathcal{L}^x$ and $\mathcal{L}^y$.
5. We use the full model but disable the repetition loss term $\mathcal{L}^c$.
6. We use the full model but disable the relative bias terms.
7. We apply the full model with no modification.

All configurations are run with the same settings mentioned in Section 4.1.3. The results are listed in Table 4.5. Predictably, using text embeddings alone underperforms all alternatives. Adding raw feature vectors alone boosts semantic labeling F1 by 7.6% and link prediction F1 by 78.17%. Notably, the embeddings trained using MLLM underperform compared to raw feature vectors. This indicates that MLLM may not be a sufficiently instructive objective for learning about layouts.

Rows 4, 5 display ablation results over metric-learning losses, namely alignment losses $\mathcal{L}^x$ and $\mathcal{L}^y$, and repetition loss $\mathcal{L}^c$. Alignment and repetition losses (particularly the alignment loss) seem to contribute to the semantic labeling task more than the linking task.

In contrast, as shown on row 6, the removal of relative bias terms has a larger impact on link detection than on semantic labeling. This is consistent with the intuitive interpretation of relative bias terms as parameters that capture contrast. AS previously mentioned, contrast is a key indicator of correspondence (and hence linkage).

Overall, the biggest jump in performance occurs between the third and fourth rows, in other words between the MLLM configuration and the ablated versions of the full model, indicating that the expressive features and metric learning paradigm contribute to the model's performance on both tasks. In future studies we hope to enhance the expressivity of layout tokens by adopting more sophisticated metric learning methods.

4.1.5 Conclusion

In this section, we presented an approach to modeling layout that encodes essential aspects of layout design. The efficiency of the representations renders them useful in cases where computational resources are scarce. On the semantic labeling task, APReCoT performed comparably to graph-based SOTA models. We also showed competitive performance on the challenging link prediction task and exceeded the performance of LayoutLM with 35x more parameters.

The assumptions about the layout of a document are often not independent from its domain and language. Assumptions such as left-to-right or top-to-bottom order, the orientation of text segments, and the relative size and placement of components can limit the application of layout-

[7] Since feature vectors are not bounded by a dictionary, reconstruction loss cannot be applied as classification loss. We tested two alternatives: 1) creating a pseudo-dictionary based on feature vectors observed during training and using cross entropy loss, or 2) using a cosine embedding loss. Both underperformed compared to MSE loss.

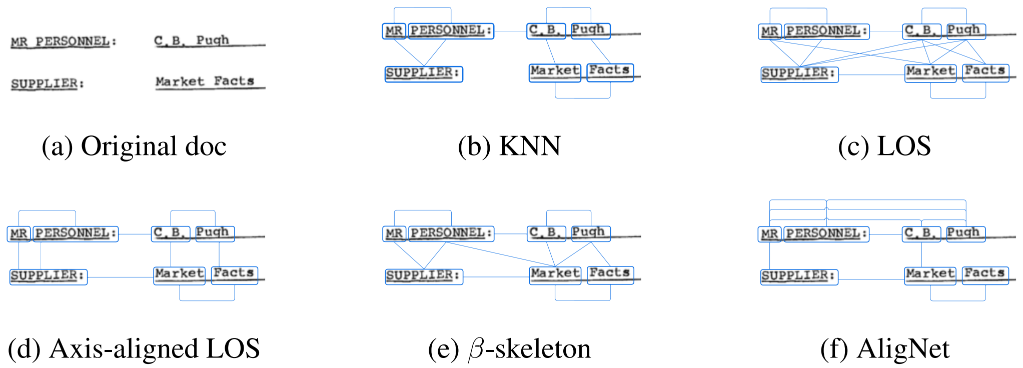
Model	Semantic labeling			Link prediction		
	P	R	F1	P	R	F1
Text only	57.68	61.43	59.50	20.11	20.87	20.48
Feature vectors	63.74	64.26	64.00	36.00	36.99	36.49
MLLM	61.49	61.46	61.47	30.37	35.44	32.71
No alignment loss	67.24	67.97	67.60	51.95	57.16	54.43
No repetition loss	69.05	69.06	69.05	51.96	57.28	54.49
No relative bias	69.85	69.64	69.74	51.58	56.51	53.93
Full model	**70.42**	**71.13**	**70.77**	**52.07**	**57.32**	**54.57**

All results reflect best performance over 100 runs.

Table 4.5: Ablation results for various pre-training settings. From top to bottom row: 1) Using BERT embeddings alone. 2) Adding raw layout features. 3) Training layout embeddings using MLLM. 4) Alignment loss disabled. 5) Repetition loss disabled. 6) Relative biases disabled. 7) The full model. The results are reported for the FUNSD dataset.

aware models to non-English documents. We hope that wider research in this area can expand the applicability and generalizability of layout-aware models.

4.2 Grounded layout generation for multimodal form understanding



(a) Original doc (b) KNN (c) LOS

(d) Axis-aligned LOS (e) β-skeleton (f) AligNet

Figure 4.4: Different graph representations for a given form.

As previously discussed, the two common tasks in form understanding include Key Information Extraction (KIE) and Relation Extraction (RE), both of which rely on identifying field names and field values, understanding tabular structures, and distinguishing between headings, main content, and other components, all of which require joint reasoning over the spatial and textual signal on each page.

With a few exceptions, most SotA researches focus on the task of KIE, disregarding RE, whereas in most applications KIE and RE need to be paired in order to identify semantically valid key-value pairs from forms. Without RE, key structural information about the document will not be captured [110, 222], and open-ended key-value extraction will be difficult [110]. Despite

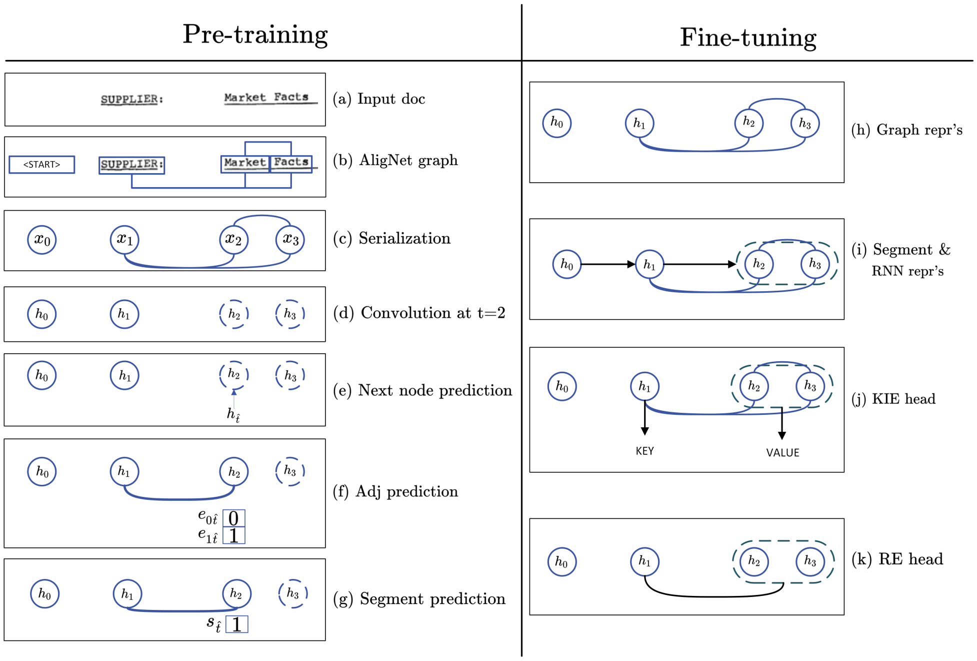


Figure 4.5: Pre-training and fine-tuning steps in our proposed approach. (a) During pre-training, a form is fed into the model as a set of tokens and bounding boxes. (b) The form is represented as an AligNet graph. (c) The serializer orders the nodes and each node is represented using its text embedding. (d) At step $t = 2$, the graph convolution produces representations $h_0 \cdots h_3$, where any edges adjacent to nodes x_2 and x_3 has been dropped, hence masking their spatial information. (e) The next node predictor uses a pointer mechanism to correctly predict the next node representation to be h_2. (f) The model predicts the adjacency vector between x_2 and the previous nodes as a binary vector. (g) The model predicts a segmentation flag for x_2. (h) During fine-tuning, AliGATr generates graph representations for each node. (i) Using the model's segmentation flags, the graph is split into segments, and an RNN is used to create sequence representations for each segment. (j) For the KIE task, a classification head predicts a class for each segment. (k) For the RE task, a link prediction head predicts the edges between segments.

this, RE remains underexplored in the form understanding literature [50, 99, 110], posing major challenges to downstream applications.

Furthermore, most models require extensive pre-training data and infrastructure to perform at the SotA level. As an example, the most popular pre-training dataset is the IIT-CDIP dataset [91], composed of 11 million images. Lastly, the trade-off between grounding (i.e. providing bounding boxes for each output token such that it can be traced back to the input) and calibration (i.e. producing distributionally robust probabilities) is difficult to balance. Small, efficient, robust, and well-calibrated models remain difficult to obtain for users with limited access to large-scale pre-training data or compute.

In this Section, we introduce **AliGATr**, a new form understanding model that addresses the above challenges by combining a graph-based representation and a layout-generation objective. By focusing on the generation of layout (as opposed to the joint generation of text and layout), our approach leads to a model that is more compact and converges using a smaller pre-training dataset. Our proposed graph representation, which we name **AligNet**, enables the model to cover both KIE and RE tasks, leading to SotA performance on the former and exceeding SotA performance on the latter. Even though AliGATr has a generative objective, it samples its output from input tokens, leading to logits that are well-grounded and well-calibrated.

Concretely, our study offers the following contributions to the literature on visually rich form understanding (VrFU):

- We propose AligNet, a graph representation technique for form documents, inspired by the four principles of layout design [75]. AligNet uses soft alignments between tokens to capture short- and long-range spatial dependencies. Its alignment-based structure (compared to the proximity-based structures often used in SotA models) allows it to propagate information more effectively.
- We introduce AliGATr, a GNN-based method inspired by GraphRNN [214], which uses a generative objective to learn layout-aware node representations. The generative objective combines next-node selection with adjacency prediction, allowing the model to recreate the layout of a page token by token. To the best of our knowledge, AliGATr is the first graph-based model to use a generative objective for form understanding.
- With 30% fewer parameters compared to the smallest SotA baseline, and using a small pre-training dataset of 1 million documents, AliGATr performs competitively on the KIE and RE tasks. Furthermore, we show that our model produces better-calibrated output distributions compared to baselines and is not over-confident.

4.2.1 Background

Research in visually rich document understanding has explored models in two architectural paradigms, namely transformer-based models and graph-based models.

Transformer-based models

Transformer-based models such as BROS [50], Docformer [9], and the LayoutLM series [56, 206, 207] are often inspired by encoder-only architectures and use an adaptation of Masked Language Modeling (MLM) [36] such as Masked Visual Language Modeling [99, 206, 207], Masked Sequence Modeling [42], learning to reconstruct [9], word-patch alignment [56], and vision-language alignment [42]. A drawback of encoder-based models is that their output probabilities aren't well calibrated [82]. This means that the output probabilities of these models don't reflect their performance, as the models can be arbitrarily over- or under-confident [67].

In recent years, the adaptation of autoregressive language models to the task of document understanding has produced models that favor a decoder-based architecture and follow generative objectives such as next word prediction [171] or block infilling [189]. While often better calibrated, these models sample their output from the vocabulary (as opposed to the input) and are therefore not guaranteed to produce outputs that can be grounded within the input. This is important for information extraction tasks, where the output should be traceable back to the input (see Section 1.1).

Another challenge of transformer-based models is their performance on associative tasks. Proper understanding of a form relies on two tasks–the extractive task of KIE, and the associative task of RE. Transformer-based models have consistently underperformed on RE compared to graph-based models such as VisualFudge [31], or hybrid graph-transformer models such as GeoLayoutLM [110] and RE^2 [147].

Graph-based models

Through their topology, graphs provide a natural way to encode the grid structure of form documents and allow more control over how information propagates across the nodes.

The graph representation in SotA studies captures each token on the page as a node, and the adjacency structure often follows one of the below paradigms [192]:

In **KNN** graphs, each node is connected to its K closest neighbors on the page (see Figure 4.4b). Due to the dependency on the parameter K, it is difficult to guarantee optimal density (or optimal sparsity) throughout the graph.

Line of Sight (LOS) graphs connect each node to other nodes within its "line of sight" (see Figure 4.4c). This guarantees that nodes that are adjacent on the page are connected, but LOS graphs can still introduce edges that don't carry meaningful information., e.g. the connection between "SUPPLIER" and "Pugh" in Figure 4.4c.

The **β-skeleton** graph can be thought of as a "ball-of-sight" approach [192] that removes some of the edges from LOS by favoring proximity (see Figure 4.4e). This approach has been adopted in line and paragraph-detection models [106, 192] as well as form extraction models [87, 89].

As can be seen in Figure 4.4e, even though the β-skeleton graph captures more meaningful relationships compared to KNN and LOS graphs, it can still produce unhelpful edges, e.g. the edge between "PERSONNEL" and "Market". This is because, like KNN and LOS, β-skeleton graphs favor proximity over alignment, whereas alignment is not only one of the core principles of layout design, but is crucial to maintaining the grid structure in forms [75]. An alternative to LOS, namely Axis-aligned LOS (Figure 4.4d) has been proposed to capture alignments, but as Davis et al. [31] argued, it is not effective for form understanding tasks due to its over-sparsity.

As shown by studies such as Liu et al. [106], the β-skeleton graph can be enhanced by the addition of redundant (or "multi-hop") edges. We adapt this idea to the Axis-aligned LOS structure, and propose a new graph structure which we name AligNet (see Figure 4.4f).

AligNet captures short- and long-range dependencies by adding multi-hop edges to the Axis-aligned LOS structure, which helps the graph honor alignment as well as proximity in modeling the layout of a page. We demonstrate AligNet's ability to capture the global structure of each page using a community detection method. Additionally, when equipped with a graph convolution network, the AligNet structure can route messages between nodes that are meaningfully associated, such as field names and field values. Our proposed graph learning approach, AliGATr, couples the AligNet representation with a layout generation objective, which leads to competitive performance on VrFU tasks, including key information extraction and relation extraction. To balance the calibration and grounding tradeoff, AliGATr uses a generative architecture, but uses a Pointer mechanism [158] to strictly produce output tokens that are extracted from the input. Furthermore, because of its generative objective, AliGATr's logits are better calibrated than encoder-based models.

In summary, AliGATr addresses the previously mentioned shortcomings of SotA approaches using the below solutions: 1) Lack of attention to alignments is resolved using an alignment-based structure (i.e. AligNet). 2) Over-sparsity of alignment-based structures is addressed by the introduction of redundant edges in AligNet. 3) Poor calibration is addressed by following a generative objective. 4) Poor grounding is addressed by using a Pointer mechanism.

The following sections present our methodology and experimental results.

4.2.2 Methodology

In this section, we describe our proposed graph representation for documents (AligNet), as well as our proposed model architecture (AliGATr). Figure 4.5 shows the overall flow of pre-training and fine-tuning steps.

AligNet

We model each document as an undirected graph $G = (V, E)$, where each node x_i represents a token on the page, and two nodes x_i and x_j are adjacent if their bounding boxes are horizontally or vertically aligned[8]. We define alignment between x_i and x_j as $\exists c \in \{\mathrm{left}, \mathrm{center}, \mathrm{right}, \mathrm{top}, \mathrm{middle}, \mathrm{bottom}\} : |b_i^c - b_j^c| < \mathcal{D}$, where b_i^c represents the coordinates of the bounding box of x_i, and $\mathcal{D}$ is a threshold that is expressed as a percentage of page width/height and can be tuned as a hyperparameter[9]. Figure 4.5(a) shows a small snippet of a form. In 4.5(b), the form has been converted into an AligNet graph (see Figure B.4b for a more substantive example).

We represent each node x_i by it embedding vector $\mathbf{x}_i$ that is generated by a language model such as RoBERTa [107].

An edge between x_i and x_j is represented by the below attribute vector:

$$\mathbf{e}_{i,j} = [-|b_i^{\mathrm{left}} - b_j^{\mathrm{left}}|, -|b_i^{\mathrm{right}} - b_j^{\mathrm{right}}|, -|b_i^{\mathrm{top}} - b_j^{\mathrm{top}}|, -|b_i^{\mathrm{bottom}} - b_j^{\mathrm{bottom}}|,$$
$$b_i^{\mathrm{height}} - b_j^{\mathrm{height}}, \frac{b_i^{\mathrm{width}}}{\mathrm{numchars}(x_i)} - \frac{b_j^{\mathrm{width}}}{\mathrm{numchars}(x_j)}]$$

Note that the first four elements show the negative absolute distance (i.e. proximity) between the four coordinates of the bounding boxes[10]. The fifth element shows the difference in the heights of the two bounding boxes, and the last element shows the difference in their average width per character. In order to avoid the need to resample all images to be of the same size, we normalize all coordinates based on the width and height of each page.

[8] Any methodology that relies on alignment-based signal, such as the AligNet structure, is at risk of failing to recognize noisy alignments, e.g. on skewed or tilted pages. We rely on the accuracy of OCR software to recognize the angle at which the document is presented, which may not always be reliable. However, as with segment/line detection, the rotation detection capability of modern OCR software has substantially improved.

[9] Alternatively, alignments can be found using more sophisticated clustering-based or convolutional methods, but based on our experiments, the simple threshold-based approach yields comparable performance.

[10] Using the negative distance is a naive but effective way to represent proximity. Our experiments demonstrated that other methods such as using the reciprocal or log-order of distance were not as effective. See Appendix B.5 for more details.

In addition to edge attributes, we also assign a label to each edge, which reflects one of the 6 possible types of alignment between the adjacent nodes, namely: left, center, right, top, middle, or bottom-aligned. The label also reflects whether the source node is located "before" the target node in the reading order, i.e. whether the source node is to the left or top of the target node. This yield 12 possible classes. In Figure 4.5(b), the edge between "Market" and "Facts" represents two directed edges: a `bottom-before` edge from "Market" to "Facts", and a `bottom-after` edge from "Facts" to "Market". This means that "Market" and "Facts" are bottom-aligned and "Market" comes before "Facts".

AliGATr

The AliGATr architecture is composed of three modules, inspired by Lee et al. [89]: a serializer, a GCN, and a decoder. During pre-training, the serializer arranges the nodes into a sequence, and the GCN generates node embeddings using generative objectives. During fine-tuning, the decoder predicts node labels (KIE) or links (RE).

Serialization The serializer uses a simple heuristic to order the tokens in a sequence. Using the top-left coordinates of each bounding box, the serializer orders the tokens in a left-to-right and top-to-bottom sequence. For English-language documents, this is meant to mimic reading order, even though it is a noisy approximation.[11] Figure 4.5(c) illustrates the serialized graph for the example in 4.5(b). This serialized sequence is used to traverse the AligNet graph during pre-training, as described in the next section.

Generative pre-training After the serializer determines the ordering of nodes, the AligNet graph is fed into a GCN. We use a Relational Graph Attention Network (RGAT) [17] as the GCN backbone. The model follows an auto-regressive layout-generation objective coupled with a segmentation objective. We describe these objectives below.

Layout generation objectives: First, we add a dummy start node x_0 to the graph that is not connected to any other nodes. This node functions as the `<start>` token for our generative task. At each timestamp t, the model masks the bounding box coordinates of nodes $x_t, x_{t+1}, \cdots, x_T$ as well as any edges adjacent to them. The model then generates representations for nodes $x_0, x_1, \cdots, x_T$, namely, $\mathbf{h}_0, \mathbf{h}_1 \cdots, \mathbf{h}_T \in \mathbb{R}^d$ where d is the hidden dimension. Figure 4.5(d) shows the node representations at $t = 2$ for the example graph. Nodes with dashed borders have their bounding boxes masked and edges removed. Using these representations, the model optimizes two objectives: 1) Given $\mathbf{h}_{0:t-1}$, the model "picks" the next node $\mathbf{h}_t$ from the set of remaining nodes $\mathbf{h}_{t:T}$, where the ordering is determined by the serializer, and all positional information (i.e. bounding box coordinates) are masked for $\mathbf{h}_{t:T}$. 2) Given the predicted next node $\mathbf{h}_{\hat{t}}$, the model predicts the edges between $\mathbf{h}_{\hat{t}}$ and the subgraph composed of $\mathbf{h}_{0:t-1}$. This is akin to presenting the tokens in a random order to the model, and encouraging the model to put the layout back together token by token, placing each new token in its proper position with regards to previous tokens on the page. Since the model has access to the token identities, it

[11] There are many cases where this heuristic won't work, e.g. on multi-column pages. However, using a noisy heuristic yields a more robust model, as it prevents the model from leveraging exact reading order information [217].

is not performing *text* generation, but *layout* generation. This allows the model to learn layout-aware representations without having to fulfill the text generation objective, which is a data- and parameter-intensive task. Below, we describe the model's two objectives:

First, given the node embeddings $\mathbf{h}_0 \cdots \mathbf{h}_{t-1}$, we use a pointer mechanism [158] to select the next node from the set of remaining nodes. The pointer is implemented as scaled dot-product attention between the sequence embedding $\mathbf{h}_{0:t-1}$ and the remaining embeddings $\mathbf{h}_{t:T}$. The node whose embedding has the highest attention score is predicted as the next node:

$$\begin{aligned}
&\mathbf{h}^{(0:t-1)} = W^{(1)}\mathbf{h}_{0:t-1}^{\top}, \mathbf{h}^{(t:T)} = W^{(2)}\mathbf{h}_{t:T}^{\top} \\
&\alpha_{\hat{t}} = \text{softmax}(\frac{(\sum_{k=0}^{t-1}\mathbf{h}_k^{(0:t-1)})\mathbf{h}^{(t:T)}}{\sqrt{d}}) \\
&j = \arg\max_i\{\alpha_{\hat{t},i}; i \in \{0, 1, \cdots, T-t-1\}\} \\
&\mathbf{h}_{\hat{t}} = \mathbf{h}_{j+t+1}
\end{aligned}$$

where $W^{(1)}$ and $W^{(2)} \in \mathbb{R}^{d\times d}$ are weight matrices, $\alpha_{\hat{t}}$ are the attention weights, $j+t+1$ is the index of the node with the highest attention weight, and $\mathbf{h}_{\hat{t}}$ is the representation of that node. In Figure 4.5(e), the next node is correctly picked as $\mathbf{h}_2$.

To calculate the next node prediction loss, we follow See et al. [158] and use Negative Log Likelihood as pointer loss: $\mathcal{L}_t^{\text{NODE}} = \frac{-\log\alpha_t}{\log(T-t)}$, where α_t is the attention weight of the correct next node x_t, and the $1/\log(T-t)$ factor is used to lower the penalty for cases when the selection of the next node has a higher degree of freedom and is therefore a more difficult task.

Once the next node $\mathbf{h}_{\hat{t}}$ is determined, the model predicts its adjacencies to the subgraph composed of previous nodes. Inspired by You et al. [214] we model this task as predicting the adjacency vector $a_{\hat{t}}$, which is a binary vector of size t where $a_{\hat{t},k} = 1$ if x_k and $x_{\hat{t}}$ are adjacent, and $a_{\hat{t},k} = 0$ otherwise. The model predicts $a_{\hat{t}}$ based on the attention between $\mathbf{h}_{0:t-1}$ and $\mathbf{h}_{\hat{t}}$. The adjacency loss is calculated using the binary cross entropy between the predicted adjacency vector $a_{\hat{t}}$ and the true vector a_t:

$$\begin{aligned}
&\mathbf{h}'^{(0:t-1)} = W^{(3)}\mathbf{h}_{0:t-1}^{\top}, a_{\hat{t}} = \frac{\mathbf{h}_{\hat{t}}^{\top}\mathbf{h}'^{(0:t-1)}}{\sqrt{d}} \\
&\mathcal{L}_t^{\text{ADJ}} = \text{BCE}(a_{\hat{t}}, a_t)
\end{aligned}$$

where $W^{(3)} \in \mathbb{R}^{d\times d}$ is a weight matrix. In Figure 4.5(f), the adjacency vector of x_2 is predicted as $[0, 1]$ indicating that no edge exists between x_0 and x_2, but an edge exists between x_1 and x_2. Note that this adjacency vector only determines the existence of an edge, without sensitivity to directionality. Directionality is only reflected in the attributes and labels of the unmasked edges.

Segmentation objective: In addition to the node and adjacency prediction objectives, the model predicts the boundaries of various segments on the page. This is to encourage the model to find groupings of tokens that correspond to an entity. Most OCR engines provide segment boundaries based on the spacing between the tokens on the page. The model uses this information to predict whether a given token marks the beginning of a new segment[12]. The loss is modeled as a simple binary cross entropy: $\mathcal{L}_t^{\text{SEG}} = -(s_t \log s_{\hat{t}} + (1-s_t)\log(1-s_{\hat{t}}))$,

[12] Studies such as Huang et al. [56] and Luo et al. [110] also use segment-level information. Note that AliGATr only uses segment-level information at training time and does not expect this information during inference.

where $s_{\hat{t}} = w^{(4)}\mathbf{h}_{\hat{t}}^{\top}$ is the predicted binary segmentation flag for $x_{\hat{t}}$, s_t is the true flag, and $w^{(4)} \in \mathbb{R}^{1\times d}$ is a weight vector. Figure 4.5(g) shows that the segmentation flag for x_2 has been predicted as 1, indicating that it signals the start of a new segment.

The total pretraining loss at step t is calculated as the sum of node prediction, adjacency prediction, and segment boundary prediction losses: $\mathcal{L}_t = \mathcal{L}_t^{\text{NODE}} + \mathcal{L}_t^{\text{ADJ}} + \mathcal{L}_t^{\text{SEG}}$.

AliGATr uses these objectives to learn layout-aware representations for each node x_t.

Fine-tuning At fine-tuning time, we use the segmentation flags learned by the model to identify the boundaries of each entity. This reduces the complexity of the downstream KIE and RE tasks since they both rely on entity-grouping to produce accurate output. Figures 4.5 (h) and (i) show the pre-segmentation and post-segmentation stages of the example graph, respectively.

We implement two fine-tuning heads, each corresponding to one of the two target tasks, i.e. KIE and RE. KIE from forms can be modeled as a node classification problem, and RE can be modeled as a link prediction problem.

The KIE node classification head uses the ordering created by the serializer to generate a sequence representation using an RNN [49] (Figure 4.5(i)). The sequence can then be used to predict I-O-B tags for each token. Finally, the KIE classification loss, $\mathcal{L}^{\text{CLF}}$, can be calculated as cross entropy loss between the predicted and true classes. The introduction of the RNN is important as it models the sequentiality of the input more effectively than the graph. However, if its representations deviate too much from the those created by the graph, they can "unlearn" certain semantic information. Inspired by Yao et al. [208], we introduce an auxiliary co-distillation loss that keeps the RNN representations ($\mathbf{h}^{\text{RNN}}$) and the graph representations ($\mathbf{h}^{\text{GNN}}$) close to each other:

$$
\begin{aligned}
\mathcal{L}^{\text{CoD}} &= \frac{1}{N}\sum_{i=1}^{N} \text{CL}(\mathbf{h}_i^{\text{GNN}}, \tilde{\mathbf{h}}_i^{\text{RNN}}) + \text{CL}(\mathbf{h}_i^{\text{RNN}}, \tilde{\mathbf{h}}_i^{\text{GNN}}) \\
\mathcal{L}^{\text{KIE}} &= \mathcal{L}^{\text{CLF}} + \mathcal{L}^{\text{KIE}} + \mathcal{L}^{\text{SEC}}
\end{aligned}
$$

where CL stands for the contrastive loss described in Tian et al. [174] and $\tilde{\cdot}$ is the stop-gradient operator, which freezes the corresponding representation. The model continues to learn segmentation during fine-tuning via the segmentation loss $\mathcal{L}^{\text{SEG}}$. Figure 4.5(j) shows the final output of the KIE classification head.

The RE link prediction head does not require serialization, as it simply uses the dot product of two node representations $\mathbf{h}_i^{\text{GNN}}$ and $\mathbf{h}_j^{\text{GNN}}$ to predict whether an edge exists between them. The RE loss, $\mathcal{L}^{\text{RE}}$ is calculated based on the binary cross entropy between predicted and true edges. Figure 4.5 (k) shows the output of the RE head. Note that the RE head would be able to identify relations between nodes and segments, even if they are not aligned. The alignment edges are only used during pre-training to create layout-aware node representations. See Figure B.2 for a examples of unaligned RE results.

4.2.3 Experiments

In this section we describe the datasets and baselines used in our experiments. Other experimental settings are described in Appendix B.1.

Dataset	# Train	# Test	Tasks	# Classes
FUNSD [63]	149	50	KIE, RE	4
SROIE [57]	626	347	KIE	4
CORD [128]	800	100	KIE, RE	30
BuDDIE [233]	1,172	332	KIE	69

Table 4.6: Statistics about four datasets that cover KIE and RE tasks. Note that we only list the tasks that are used in our experiments. "# Classes" indicates the number of entity classes used in the KIE task.

Datasets

We use four multimodal form understanding datasets that cover KIE and RE tasks. **CORD** and **SROIE** are collections of retail receipts. **FUNSD** includes research and advertising forms sampled from the RVL-CDIP dataset [45], and **BuDDIE** is a collection of business entity filings collected from various US states. Table 4.6 shows high-level statistics about each dataset.

Baselines

We use four SotA baselines in multimodal form understanding. **LayoutLMv3** [56] is a transformer-based model that uses vision, spatial, and text signal to model multimodal documents. By abandoning a complex Region-Proposal Network in favor of a simple patch-based vision encoder, LayoutLMv3 reduces the number of parameters compared to LayoutLMv2 [206], while achieving superior performance on the KIE task[13]. **GraphLayoutLM** [96] enhances LayoutLMv3 with a graph component that maps the relative positioning of various nodes with regards to each other, improving performance on KIE. **GeoLayoutLM** [110] adds geometric constraints to LayoutLMv3 and demonstrates SotA performance on both the KIE and RE tasks. Lastly, **FormNetv2**[14] [89] uses a β-skeleton graph and a Graph Convolution Network to model visually rich forms. In contrast to previous models, FormNetv2 does not rely on segment-level bounding boxes, and relies entirely on token-level presentations. The model outperforms LayoutLMv3 on KIE despite a 44% reduction in model size.

4.2.4 Results and discussion

Performance on KIE and RE tasks

Table 4.7 shows the performance of AliGATr and four baselines on the multimodal form datasets. As mentioned in Section 4.2.3 three of the four baseline models rely on segment-level bounding boxes, while FormNetv2 and AliGATr do not rely on segment-level bounding boxes during inference, and only use token-level bounding boxes. To make the comparisons consistent across all models, we have reported the performances using token as well as segment bounding boxes

[13] We do not cover the performance on tasks that are out of scope for AliGATr, such as document classification and visual question answering

[14] Note that we do not include Multimodal LLMs such as UReader [209] or DocLLM [189] because they have not yet achieved SotA performance on form processing tasks. OCR-free models such as UDOP [171] and mPLUG-DocOwl1.5 [52] are excluded for the same reason.

Model	Modalities	# Params	Pre-training dataset size	FUNSD	CORD	SROIE	BuDDIE[15]
LayoutLMv3LARGE	T+L+I	357M	11M	82.53/92.08	95.92/97.46	94.96/98.63	83.42
GraphLayoutLMLARGE[16]	T+L+I	372M	11M	-/**94.39**	-/97.75	-/-	-
GeoLayoutLM	T+L+I	399M	11M	84.40/92.86	96.57/97.71	95.04/98.70	**84.86**
FormNetv2[17]	T+L+I	204M	11M	**86.35**/92.51	97.37/97.70	98.31/-	-
AliGATr	T+L	145M	1M	86.31/92.95	**97.48/97.83**	**98.57/98.78**	81.85

Table 4.7: Performance on the KIE task. "T", "L", and "I" stand for text, layout, and image. The performance is reported as `token/segment`, where `segment` indicates performance when segment-level bounding boxes are available at test time, and `token` indicates performance when only token-level bounding boxes are available.

(see caption for more detail). Despite a 30% reduction in size compared to the smallest baseline (FormNetv2), AliGATr performs on par with or better than the SotA models on the KIE task. The model falls short of SotA on BuDDIE, which has the largest number of classes and is composed of denser documents (business entity filings).

Table 4.8 shows the performance of AliGATr and two other baselines on the RE task. Once again, AliGATr matches or outperforms SotA models despite having 60% fewer parameters than the smaller baseline (LayoutLMv3LARGE).

Model	Modalities	# Params	FUNSD	CORD
LayoutLMv3LARGE	T+L+I	357M	80.35	99.64
GeoLayoutLM	T+L+I	399M	89.45	**100.00**
AliGATr	T+L	145M	**89.50**	**100.00**

Table 4.8: Performance on the RE task. The numbers reported for the CORD dataset correspond to the "REaKV" task mentioned in Luo et al. [110].

Calibration

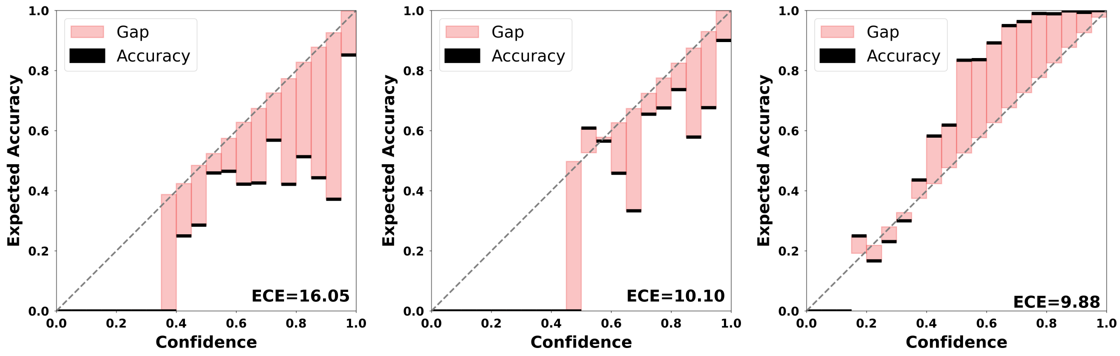
There are two aspects of calibration that facilitate straight-through-processing of documents in downstream applications. The first is the confidence of the model with regards to the output. Under-confidence and over-confidence are both problematic as they do not reflect the model's true performance. The second, and arguably more important aspect is the consistency of the confidence gap. If a model is consistently over or under-confident, it is much easier to set a fixed threshold beyond which the model's outputs can be trusted.

Figure 4.6 shows the confidence versus performance plot for LayoutLMv3LARGE, GeoLayoutLM, and AliGATr, when finetuned on the FUNSD dataset. As the Figure shows, AliGATr's output probabilities are better calibrated, and do not exhibit the over-confident trend that is observed in the baselines. As indicated by the lower ECE, AliGATr is also more consistent in its confidence gap, and a confidence threshold of 0.8 and above yields near perfect performance.

[15] The BuDDIE dataset does not provide segment level bboxes. Therefore only token-level results are reported.

[16] Some experimental results are missing for GraphLayoutLM because the authors were not able to recreate the baselines reported by Li et al. [96], and are therefore only reporting the numbers disclosed in the original paper.

[17] FormNetv2 is not Open Source. Therefore only the results reported in Lee et al. [89] are included.



(a) LayoutLMv3 (b) GeoLayoutLM (c) AliGATr

Figure 4.6: Calibration plots and ECE measures for AliGATr versus two baselines. All models have been finetuned for the KIE task on the FUNSD dataset.

4.2.5 Ablation and sensitivity studies

In this section, we investigate how three components of our proposed pipeline contribute to downstream performance. Due to infrastructure limitations, all of the studies reported here are based on a toy pre-training dataset of 30K examples sampled from OCR-IDL [13]. The gains/drops in performance are statistically significant at $p < 0.005$, based on the paired-bootstrap test proposed by Berg-Kirkpatrick et al. [12], with $b = 10^2$. Therefore we expect the trends to hold for larger pre-training datasets.

Approach		**KIE**	**RE**
Graph Structure	β-Skeleton	50.89	64.52
	AligNet	**51.30**	**73.12**
Serialization	No node prediction	50.44	70.01
	No edge labels	49.29	68.43
	Order-invariant labels	51.03	71.26
	Order-sensitive labels	**51.30**	**73.12**
Full Gen.	$N = 1$	51.30	**73.12**
Skip Gen.	$N = 5$	50.94	65.14
	$N = 10$	50.39	64.97
	$N = 20$	50.50	64.09
Chunk Gen.	$M = 20$	51.28	73.07
	$M = 50$	**51.31**	72.98
	$M = 100$	51.29	73.01

Table 4.9: The impact of graph representation, edge representation, and generation methods on the KIE and RE tasks (F1 performance on the FUNSD dataset).

Graph structure

As mentioned in Section 4.2.1, β-skeleton graphs are a common choice in graph-based models. The top segment of Table 4.9 shows the performance of the β-skeleton graph against the AligNet structure. The β-skeleton graph slightly underperforms AligNet on the KIE task, but has an even larger gap on the RE task. The latter is expected, as alignments often play a major role in indicating semantic correspondence between field names and values. This demonstrates the effectiveness of the AligNet structure in modeling form understanding tasks. For further analysis on this topic, see Appendix B.3.

Serialization

As discussed in Section 4.2.2, the serializer orders the nodes in left-to-right and top-to-bottom fashion. This has an impact on two components of AliGATr, namely the next node predictor (which is designed to predict the next node in the sequence according to the serializer's ordering), and the edge labels (which are determined based on the relative position of two nodes on the page).

The second segment of Table 4.9 shows the impact of ablating these components. Without node prediction, both KIE and RE tasks suffer. Removing edge labels has an even bigger impact on performance, even though order-invariant labels recover some of the performance. The best performance belongs to a model that has order-sensitive edge labels (i.e. 12 classes, as described in Section 4.2.2), which is therefore the model used in our final experiments. For a deeper analysis on how edge representations can impact downstream performance, see Appendix B.5.

Generation regime

Lastly, we analyze the impact of the generation regime on downstream performance. In the default auto-regressive setting, every token is generated one by one. This can be costly if the number of tokens on a page is large. SotA models such as LayoutLMv3LARGE cap the sequence length at 512 tokens which poses a risk for text-heavy pages. Furthermore during pre-training the model might not be exposed to sections that usually appear at the bottom of the page, e.g. footers or page numbers. Instead of truncating the input during pre-training, we experiment with two alternatives. In **Skip Generation**, the model generates every N tokens. In **Chunk Generation**, the model generates a randomly sampled subsequence of length M from each page. The last segment of Table 4.9 shows the model's performance in these settings. In the default setting (titled "Full Gen."), the model has the highest performance on RE and close to highest performance on KIE. The performance suffers when switching to Skip Generation, especially for RE. This may be attributed to the disjointedness of generations, because skipping over $N - 1$ tokens can obscure the relationship between neighboring tokens. This problem is largely addressed by Chunk Generation, as is evident from the model's performance, even when M is small. Given the competitive performance of Chunk Generation with $M = 20$, we selected this setting to perform pre-training. A possible risk of Chunk Generation is that the robustness of output probabilities might be undermined, but, as presented in Section 4.2.4, the model has better calibrated output than baselines. The effectiveness of Chunk Generation further demonstrates the robustness and efficiency of AliGATr's learning objectives.

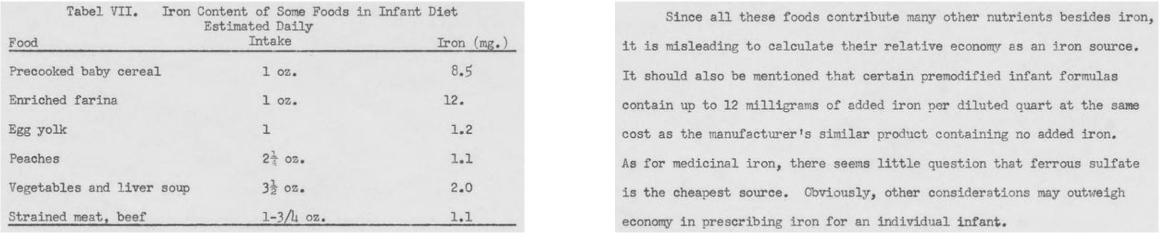
4.2.6 Conclusion

In this Section, we presented AliGATr, a layout generation technique for form understanding that is competitive with SotA on Key Information Extraction and Relation Extraction tasks, using 30% fewer parameters and 11x fewer training examples. We showed how, despite using the spatial and textual modalities alone, and relying on subsequence generation, the model produces better-calibrated probabilities. In future studies, we hope to investigate AliGATr's effectiveness in other adjacent tasks that lend themselves to graph-based representations, such as document classification, page segmentation, and structure extraction.

Chapter 5

Grounded evaluation

Visual Question Answering (VQA) over multimodal documents requires joint reasoning over textual, spatial, and visual signals. Several benchmarks have been proposed to measure the performance of SotA models on this task, including single-page and multi-page VQA [113, 114, 175, 176, 179]. In these benchmarks, the ground truth answer is expressed as a sequence of tokens, and evaluated against the sequence of tokens produced by each model. As such, the evaluation metrics that these benchmarks employ are focused on the surface similarity between the model output and the ground truth answer. This misses two key aspects of the model's output: 1) Is it aligned with the expected semantic category? For example, if the ground truth is a number, is the model also producing a number (or a related expression)? 2) Can it be located within the input document? In other words, is the model hallucinating a response, or is it generating something based on the document (even if it's wrong)? Grounded responses help determine the provenance of the model's output and verify its accuracy.



Tabel VII. Iron Content of Some Foods in Infant Diet

Food	Estimated Daily Intake	Iron (mg.)
Precooked baby cereal	1 oz.	8.5
Enriched farina	1 oz.	12.
Egg yolk	1	1.2
Peaches	2¼ oz.	1.1
Vegetables and liver soup	3½ oz.	2.0
Strained meat, beef	1-3/4 oz.	1.1

(a) Tabular snippet.

Since all these foods contribute many other nutrients besides iron, it is misleading to calculate their relative economy as an iron source. It should also be mentioned that certain premodified infant formulas contain up to 12 milligrams of added iron per diluted quart at the same cost as the manufacturer's similar product containing no added iron. As for medicinal iron, there seems little question that ferrous sulfate is the cheapest source. Obviously, other considerations may outweigh economy in prescribing iron for an individual infant.

(b) Text snippet.

Figure 5.1: Two excerpts from an image document from the DocVQA dataset [113].

Figure 5.1 and Table 5.1 illustrate this using an example from the DocVQA benchmark [113], which uses Normalized Levenshtein Distance [86] as its evaluation metric. Given the two excerpts from an image document in Figure 5.1, two questions are listed in Table 5.1. The first question, "How many mgs of iron is in enriched farina?", requires the model to reason over a tabular structure. If the model produces "26" as the answer, it will be rewarded by a score of 0.5 because "26" shares one digit with the ground truth answer, "12". In contrast, if the model produces "8.5" as the answer, it will not be rewarded, as there is no overlap with the ground truth. This is potentially problematic, as the first answer is not mentioned anywhere on the page,

Table 5.1: Two example questions based on the snippets in Figure 5.1. The "NLS" column shows the score awarded to hypothetical answers for each question using the NLS metric [113]. In "Ours", we show how our proposed score is calculated.

Question	Context	GT Answer	Predicted Answer	NLS	SMuDGE (Ours)						
					Match Score			Grounding Score			Composite Score ($\alpha = 0.25$)
					Text Score	Num Score	Agg.	Horizontal Distance	Vertical Distance	Agg.	
How many mgs of iron is in enriched farina?	Figure 5.1a	12	26	0.5	-	0.0	0.0	-	-	0.0	0.0
			8.5	0.0	-	0.0	0.0	~ 0.0	0.2	0.02	0.01
How much added iron do premodified infant formulas contain?	Figure 5.1b	up to 12 milligrams	up to 12 mgs	0.58	0.59	1.0	0.74	0.0	0.0	1.0	0.93
			up to 1z milligrams	0.95	0.94	0.0	0.0	0.0	0.0	1.0	0.75

and can therefore be considered hallucinatory. The second answer, while inaccurate, captures a number that is present on the table in Figure 5.1a, and is located on the same column as the ground truth, potentially signifying some level of tabular reasoning by the model. A more robust evaluation metric would provide a small reward to the second answer, and give the first answer a score of 0.0.

Another question, "How much added iron do premodified infant formulas contain?", requires verbal reasoning over the paragraph in Figure 5.1b. If a model responds by "up to 12 mgs", it is penalized for the surface dissimilarity to the ground truth answer, "up to 12 milligrams". In contrast, if the model produces "up to 1z milligrams", it is awarded a higher score since its answer has a larger overlap with the ground truth. Again, this is problematic as the second answer misrecognizes a key component of the ground truth (i.e. the number), and as such indicates a completely inaccurate quantity. A more robust evaluation metric should reward a higher score to the first answer than the second.

In this Chapter, we propose a new evaluation methodology, which we name SEmantics and Document Grounded Evaluation (SMuDGE). SMuDGE addresses the above issues by grounding the similarity score in the expected output type(i.e. numeric, textual, or hybrid). We also add a new component—a multimodal grounding score that determines whether the model's output is located within the input document, and where it is located in relation to the ground truth. While grounding is a major requirement (and challenge) for the operationalization of Document VQA models [124], it is difficult to determine how much grounding might matter to one downstream application versus another. Therefore, we design our evaluation approach to accommodate different settings by allowing users to set preferred weights for each component.

Concretely, our study makes the following contributions to the field:

1. We propose a new evaluation framework (SMuDGE) that accounts for the groundedness of outputs.
2. Using SMuDGE, we re-evaluate the performance of SotA models on four common Document VQA benchmarks, and analyze the impact of grounding on the ranking of each leaderboard.
3. We perform a detailed analysis of the types of questions and answers most impacted by grounding-sensitive criteria, and propose a configurable setting that allows the downstream users of each model to tune the evaluation to their needs.

4. Our analyses show that SMuDGE produces scores better aligned with human preferences.
5. We experimentally demonstrate that better-grounded generation is associated with better calibrated outputs.
6. Lastly, our analyses show that SMuDGE rewards models that are more robust to variations in tasks and datasets.

5.1 Background

In recent years, generative multimodal models have made major strides in Visual Question Answering over image documents. As an example, as of October 2024, the top-performing model on the DocVQA leaderboard is within 2 points of human performance[1].

A key challenge of generative models is that their output is difficult to ground within the input document [232]. Since generative models produce sequences that are sampled from the vocabulary, they are not guaranteed to generate answers that are based on the input, unless forced to do so via grounded decoding (e.g. as in [137]). This in turn makes it difficult to detect hallucinations, establish the provenance of the model's generations, or measure the reliability of its outputs, all of which limit the applicability of such models in many enterprise domains [124].

This problem is compounded by the fact that most popular Document VQA benchmarks do not account for grounding in their evaluation criteria. A common metric used by these benchmarks is Average Normalized Levenshtein Similarity (ANLS), as proposed by Mathew et al. [113], which measures the similarity between the ground truth and predicted answers based on their edit similarity. If the NLS for a ground-truth/prediction pair is below a predefined threshold (typically 0.5), the score is flattened to zero, otherwise the NLS is used. The flexibility that the NLS metrics provides allows the benchmarks to handle minor errors such as character misspellings resulting from poor Optical Character Recognition, without over-penalizing the models.[2] Nevertheless, relying solely on surface similarity carries other risks for robust evaluation.

The issues mentioned above are rooted in two fundamental disadvantages of similarity-based metrics: 1) The metrics measure surface similarity, without accounting for how a small change in the characterization of an answer can impact its meaning (e.g. changing a single digit in a number can change its value by a large magnitude). 2) The metrics do not distinguish between answers that can be traced back to the input document, and those that can result from hallucination.

More recently, some studies have noted the shortcomings of common evaluation metrics in the field of multimodal document understanding, and proposed alternatives [231]. Most notably, Peer et al. [130] proposed ANLS*, a data-type-aware metric that can be used for single or multi-piece extraction and QA over documents. While it addresses many challenges of the ANLS metric, ANLS* is not designed to capture the multimodal groundedness of model outputs. In contrast, we focus on the challenge of measuring groundedness for extractive VQA over

[1] `https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1`

[2] Contrast this to a metric that relies on n-gram overlap metrics, or cosine similarity of distributed representations. Such metrics might consider "apple" and "app1e" to be very dissimilar words, given a single-character difference between them.

documents, where correct answers are guaranteed to be expressed in the input. We propose a configurable evaluation method that not only accounts for the groundedness of predictions, but also incorporates the type-aware nature of metrics, similar to ANLS*. To the best of our knowledge, this is the first study that examines the impact of groundedness in evaluating Document VQA models. The following section describes our proposed approach in detail.

5.2 Proposed methodology

To measure the impact of groundedness in Document VQA performance, we develop a composite score to rate the output of each model. To ensure that the score can be applied to all models and benchmarks, we assume access to four objects only: 1) The question. 2) The ground truth answer. 3) The answer provided by the model. 4) A dictionary of words and corresponding bounding box coordinates extracted from the input document. This dictionary can be obtained by applying any OCR tool to the document, though the quality of character recognition often differs between different providers. Most benchmarks provide this dictionary as part of their data release.

In the next two subsections, we describe how we calculate two subscores: 1) The multimodal grounding score addresses the question of whether the predicted answer can be located within the input document, and if so, where it is located with respect to the ground truth answer. 2) The type-aware surface similarity score evaluates the predicted answer based on its type, i.e. numeric, textual, or hybrid.

5.2.1 Multimodal groundedness

Given a question q_i, a ground truth answer t_i, and a predicted answer a_i, we develop a score g_i that places a_i within the originating document (composed of words $w_1, w_2, \cdots, w_N$ and corresponding bounding boxes $b_1, b_2, \cdots, b_N$) and measure its distance to t_i. We do this in two steps:

Locating a_i and t_i within the document. To locate t_i within the document, we find a continuous sequence of words $w_k, w_{k+1}, \cdots, w_{k+n}$ that matches t_i.[3] If no such segment is found (say, due to OCR errors), then we find a sequence that has the highest Normalized Levenshtein Similarity (NLS) to t_i. We name this sequence $\mathbf{w}_{t_i}$ and the corresponding bounding box $\mathbf{b}_{t_i}$, which is calculated by merging $b_k, b_{k+1}, \cdots, b_{k+n}$[4]. Similarly, we find the sequence $\mathbf{w}_{a_i}$ and corresponding bounding box $\mathbf{b}_{a_i}$ by placing a_i within the document. Note that a_i is not guaranteed to be found on the page, for instance in case of hallucinations. If we can't find a $\mathbf{w}_{a_i}$ such that NLS$(a_i, \text{concat}(\mathbf{w}_{a_i})) > 0.3$, then we define $\mathbf{b}_{a_i}$ as:

$$[\mathbf{b}_{t_i}^{\texttt{left}}, \mathbf{b}_{t_i}^{\texttt{top}}, -\texttt{width}_i - \mathbf{b}_{t_i}^{\texttt{right}}, -\texttt{height}_i - \mathbf{b}_{t_i}^{\texttt{bottom}}] \tag{5.1}$$

[3] Note that a multimodal document is a 2-D artifact, and therefore a "continuous sequence" can extend in multiple directions, depending on the reading order of the page. Most commercial OCR packages such as `Textract` segment each page based on semantic information, e.g. an address block is presented as one segment, even if it contains multiple lines. We therefore rely on the segments provided by these packages to determine continuity. In the absence of such information, a graph representation of the document can be used as a proxy. In Appendix C.1, we provide an algorithm that can be used to ground the sequence using this graph representation.

[4] See Appendix C.2.1 for additional details.

where $\mathbf{b}_{\cdot}^{\texttt{left}}, \mathbf{b}_{\cdot}^{\texttt{top}}, \mathbf{b}_{\cdot}^{\texttt{right}}, \mathbf{b}_{\cdot}^{\texttt{bottom}}$ indicate the four coordinates of the bounding box $\mathbf{b}_{\cdot}$ and $\texttt{width}_i, \texttt{height}_i$ indicate the width and height of the page, respectively. In other words, we use the bounding box of the ground-truth answer t_i and mirror its bottom right corner in the negative space. This ensures that the distance between $\mathbf{b}_{t_i}$ and $\mathbf{b}_{a_i}$ is measured as 1 (see below).

Measuring the distance. Next, we measure d_i, the distance between $\mathbf{b}_{a_i}$ and $\mathbf{b}_{t_i}$. We do this by first finding the centroid of each bounding box, and then measuring the Normalized Manhattan Distance (NMD) between the centroids. In other words:

$$
\begin{aligned}
d_i = {} & \\
& \left|\frac{\mathbf{b}_{t_i}^{\texttt{right}}}{2\times\texttt{width}_i} - \frac{\mathbf{b}_{t_i}^{\texttt{left}}}{2\times\texttt{width}_i} - \frac{\mathbf{b}_{a_i}^{\texttt{right}}}{2\times\texttt{width}_i} + \frac{\mathbf{b}_{a_i}^{\texttt{left}}}{2\times\texttt{width}_i}\right| + \\
& \left|\frac{\mathbf{b}_{t_i}^{\texttt{bottom}}}{2\times\texttt{height}_i} - \frac{\mathbf{b}_{t_i}^{\texttt{top}}}{2\times\texttt{height}_i} - \frac{\mathbf{b}_{a_i}^{\texttt{bottom}}}{2\times\texttt{height}_i} + \frac{\mathbf{b}_{a_i}^{\texttt{top}}}{2\times\texttt{height}_i}\right|
\end{aligned}
\tag{5.2}
$$

If the predicted answer a_i cannot be located within the document, the formulation presented in Equation 5.1 yields $d_i = 1$. Note that $0 \leq d_i \leq 1$.

Finally, we calculate the grounding score g_i by applying an exponential decay function to d_i: $g_i = e^{\frac{-d_i}{1-d_i}}$. Note that the score rewards cases where $\mathbf{b}_{t_i}$ and $\mathbf{b}_{a_i}$ are close, or horizontally/vertically aligned (due to lower Manhattan Distance) with the reward dropping exponentially with distance.

5.2.2 Type-aware surface similarity

To measure m_i, the surface match score between t_i and a_i, we follow the below criteria:

1. If t_i is textual[5], we use the NLS metric.
2. If t_i is numeric, we use a binary score that indicates whether the predicted answer matches the ground truth exactly. We allow some flexibility in the match, for example numbers scaled by 100, thousand, million, or billion are considered a match. This is to account for different expressions of percentages, basis points, financial metrics, etc.
3. If t_i is composed of both textual and numeric characters, we first create substrings num_{a_i}, str_{a_i}, num_{t_i}, and str_{t_i} by extracting the numeric and non-numeric characters of a_i and t_i, respectively. Next, we calculate the number-based and text-based scores for each substring according to the above criteria. The final score is a weighted harmonic mean of the two subscores: $\frac{11}{\frac{10}{\text{num\_score}_i}+\frac{1}{\text{str\_score}_i}}$.[6] Note that the model has to get the numeric part of the answer correctly to be rewarded higher.

5.2.3 Composite metric

Given the mutimodal grounding score g_i and type-aware match score m_i, we propose the following composite score parameterized by α:

[5] See Appendix C.2.2 for additional details.

[6] See Appendix C.2.3 for additional details.

$$s_i = \alpha m_i + (1 - \alpha) g_i \tag{5.3}$$

Note that $\alpha = 0$ yields the grounding score and $\alpha = 1$ yields the type-aware match score.

5.3 Experiments

5.3.1 Datasets

Given the composite score proposed in Section 5.2.3, we investigate the impact of groundedness on four prominent Document VQA benchmarks.

DocVQA [113] is a visual question-answering (VQA) dataset designed specifically for document images. It contains over 12,000 document images sourced from scanned business forms, reports, and invoices, among others. The dataset is structured with over 50,000 question-answer pairs, and questions are broken down into 9 categories, indicating the context of the correct answer (e.g. "Free_text", "Layout", "Figure/Diagram", etc.). This breakdown is not available for the text collection. Therefore we determine the type of each question using GPT-4o [1][7]. Next we remove questions in the "Yes/No" category to filter potentially abstractive questions. This results in 5,130 questions in the final dataset.

InfographicVQA. [114] is a dataset aimed at visual question answering over complex infographic documents. The dataset includes over 5,000 infographic images and over 30,000 questions that require reasoning over text, charts, and images embedded within the infographic. We filter multi-piece answers from the test collection, resulting in 3,272 samples.

MP-DocVQA [176] focuses on multi-page documents. It consists of over 46,000 question-answer pairs from 6,000 multi-page documents. We use 5,019 questions in the test set.

DUDE [179] is a document understanding dataset focused on structured documents such as forms, invoices, and tables. It includes around 5,000 documents and 41,000 question-answer pairs. We limit the test collection to single-piece extractive questions, resulting in 2,552 samples.

For each sample in each dataset, we calculate the NLS as well as the composite score, with α set to increments of 0.05 in the $[0, 1]$ range.

5.4 Analysis

Throughout most of our experiments, we set $\alpha = 0.25$, as it proves optimal based on the calibration analysis provided in Section 5.4.3. Since α is optimized on the DUDE dataset, we have not included this dataset in any of the analyses that use this optimal value for α.

5.4.1 Leaderboard analysis

We first analyze how SMuDGE can affect the rankings produced by Document VQA benchmarks. Figure 5.2 illustrates this using the top 10 models[8] on the DocVQA leaderboard. The left-

[7] Please see Appendix C.5 for details.

[8] As of September 2024.

most column of the figure shows the original ANLS-based ranking[9]. The second column shows how the ranking changes if we switch to SMuDGE with $\alpha = 0.25$. As the figure shows, human performance and QWen2-VL [190] remain stable, but all other models move by at least one position on the leaderboard. The middle segment of the figure shows how the models would rank based on the type of question. Certain question types such as "Figure/Diagram" and "Table/List" offer little volatility, but for questions that fall under "Handwritten" or "Other", the volatility is higher.[10] The middle segment of the figure also shows that some models such as SMoLA-PaLI-X [199] are better at answering questions based on "Free_text" contexts, whereas they struggle with "Table/List" questions compared to other models.

The right segment of the figure shows the rerankings broken down by the type of answer. As expected, textual answers offer the closest ranking to the original one produced by ANLS, whereas numeric and hybrid answers perturb the ranking of the leaderboard. Notably, humans remain the top performer for textual and hybrid answers, but fall behind two other models in the numeric category. This can be attributed to the human tendency to rephrase certain entities such as numbers and dates. For example, in Question #3027, the ground truth answer "(16.1%)" is rephrased as "-16.1%" by human respondents, and for question #3290, "1,700" is modified as "about 1,700".

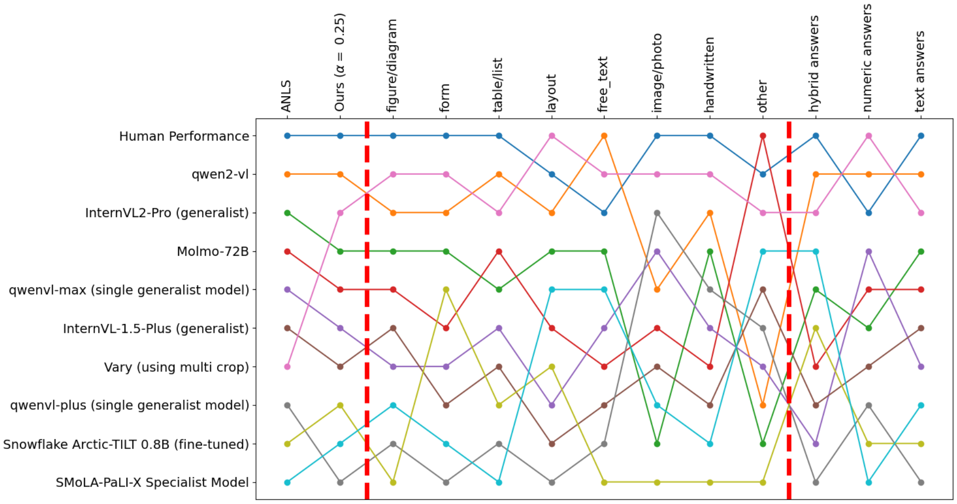


Figure 5.2: The rankings of the top 10 models on the DocVQA leaderboard, before and after applying our composite score with $\alpha = 0.25$. Left segment: Rankings based on ANLS versus our score. Middle segment: Our rankings broken down by question type. Right segment: Our rankings broken down by answer type.

Figure 5.3 shows the correlation between rankings produced by ANLS and by our composite score with $\alpha = 0.25$. Following Alzahrani et al. [5], we calculate the correlation based on a two-

[9] Note that our ANLS-based rankings could be slightly different from the leaderboard, since we have filtered the questions per Section 5.3.1.

[10] An example of a question classified as "Other" is: "What does GCC stand for?" requiring the model to infer that an acronym mentioned on one part of a page is related to an entity mentioned on a different part. This category of questions constitutes about 0.2% of the DocVQA dataset, and can be considered negligible.

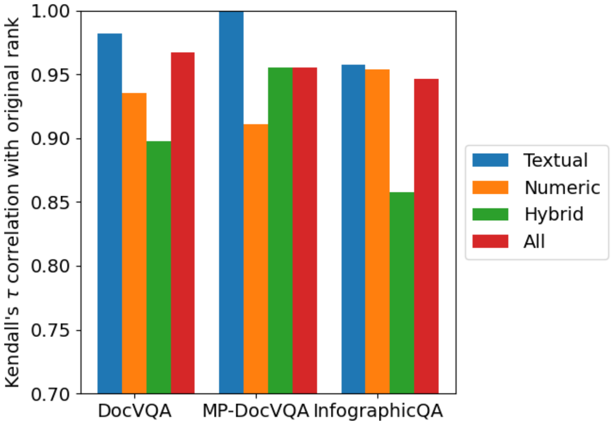


Figure 5.3: The correlation between the rankings produced by our method (with $\alpha = 0.25$) and the original ANLS-based ranking, broken down by the type of answer. All τ values are significant at $p \ll 0.05$.

tailed Kendall's τ analysis. Note that the y-axis on Figure 5.3 begins at 0.70. As the figure shows, questions with textual answers are the least affected by switching to our score, but numeric and hybrid answers impact the ranking by a larger margin. This is expected as the text-only version of our score is the closest to ANLS. Of the three benchmarks shown in the figure, InforgraphicVQA is most affected by our score, whereas DocVQA and MP-DocVQA retain a strong correlation with their original rankings. As evidenced by Figure 5.2, this strong correlation does not indicate a stable leaderboard, but one where the models move by $\pm d$, where d is a small number.

5.4.2 Question type analysis

Figure 5.4 shows the correlation between our composite score and the original ranking of the DocVQA leaderboard for each question type. As expected, moving from small values of α (weighing groundedness more that type-aware similarity) to large values (weighing type-aware similarity more than groundedness), moves the rankings closer to the original ANLS ranking. This is especially true of the "Free_text" category, where our score comes closest to ANLS. Once again, "Other" is the outlier category, which can be safely ignored due to its small sample size. The remaining categories show a similar trend, further establishing that groundedness is not accounted for in ANLS-based rankings.

5.4.3 Association with calibration

The DUDE dataset provides the confidence scores produced by each model (when available). This enables the benchmark to report Expected Calibration Errors (ECE) [126], indicating if the models are wrongfully over or under-confident about the accuracy of their output. We use this metric to determine whether our proposed score can account for accuracy through calibratedness. To do this, we map the score at various αs against the calibration error of each model, and calculate the Pearson-R correlation between the two. The results are displayed in Figure 5.5. As the figure shows, at small values of α (focusing on groundedness), there is a negative correlation

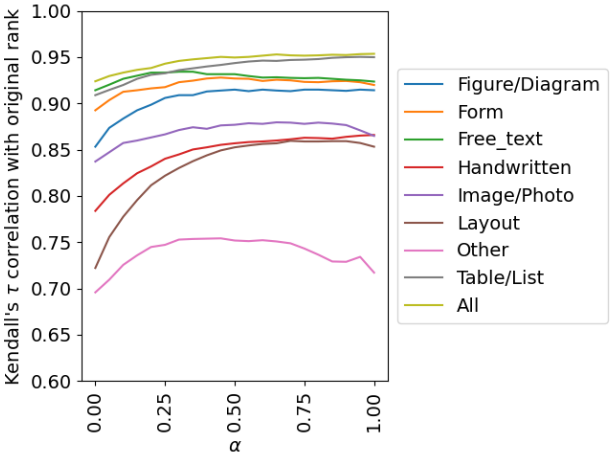


Figure 5.4: Kendall's τ rank correlation with the original DocVQA leaderboard, broken down by question types. All τ values are significant at $p \ll 0.05$.

with ECE, indicating that a higher score is correlated with a lower ECE. As α increases and the score shifts towards surface similarity, the association moves towards positive, crossing 0 around $\alpha = 0.5$. This trend can be observed for all categories of questions except "Textual" questions, which enforce surface similarity at all α values. The optimal value for α, which minimizes the correlation with ECE across most categories lies at around $\alpha = 0.25$.

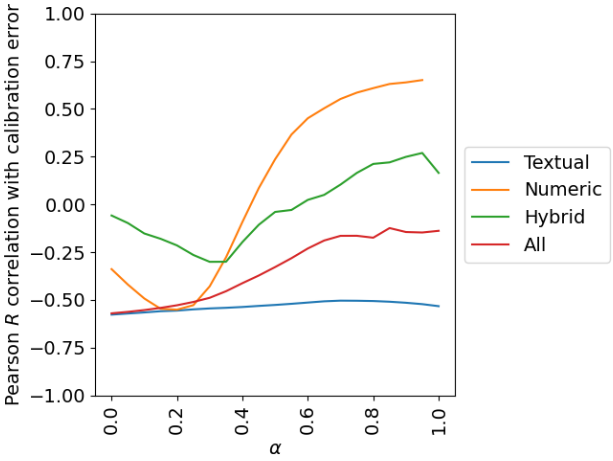


Figure 5.5: Pearson R correlation with the calibration error of models based on the DUDE leaderboard, broken down by answer type.

5.4.4 Association with robustness

Next, we inspect the association between SMuDGE and the robustness of a given model. Robustness is not a formally defined term in the Document VQA field, but can be interpreted as a model's consistent performance across different settings, benchmarks, and sample types.

Therefore, we define robustness as the volatility[11] of a model's ranking when evaluated on various subsets of questions (e.g. textual, numeric, hybrid, or all questions at once). We plot this volatility against the volatility of a model's scores, using the DocVQA, MP-DocVQA, and InfographicVQA benchmarks. Figure 5.6 shows the results using ANLS as well as SMuDGE with $\alpha = 0.25$. Each dot represents one model, with red dots representing models evaluated using ANLS, and blue dots representing models evaluated by SMuDGE. As the regression lines in the figure show, both approaches maintain a positive trend between the volatility in scores and rankings. In other words, models with stable rankings tend to have stable scores as well. However, the positive trend is stronger for our score compared to ANLS, with a small but statistically significant regression coefficient of 0.58 (compared to ANLS's 0.33).

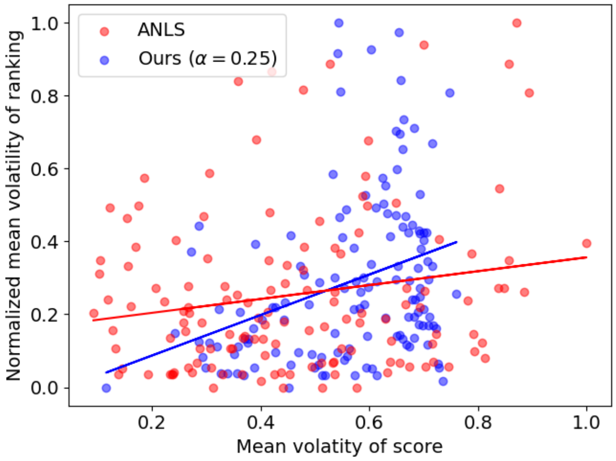


Figure 5.6: The mean volatility of each model's score versus its ranking. Red dots represent ANLS scores and blue dots represent SMuDGE with $\alpha = 0.25$.

Next, to present a qualitative view of how our score can reward robust models, we calculate a robustness score for each model in the DocVQA benchmark. To do this, we scale a model's rank volatility by its median rank. This ensures that if a model is stable across rankings, it receives a high robustness score, unless it is a generally poor performing model (e.g. a model that comes last in all rankings). Table 5.3 lists the top-5 models identified using this technique. The ANLS-based models reflect the default ranking of the DocVQA leaderboard, with Humans leading the group, followed by Large MLMs such as QWen2-VL [190] and InternVL2-Pro/InternVL-1.5 [25].

In contrast, our score produces a ranking that includes a Small MLM, namely, Arctic-TILT [15]. As of October 2024, this model is ranked 11 on the DocVQA leaderboard, above all other Small MLMs and a few Large MLMs. In addition, it is ranked 1st on the MP-DocVQA and DUDE leaderboards. No other models listed in the ANLS column show the same level of cross-benchmark robustness. Similarly, Molmo-72B [32] is 4th on the InfographicVQA benchmark. The strong cross-benchmark rankings indicate that our method can generate rankings that reward robust models.

[11] See Appendix C.2.4 for additional details.

Table 5.2: Five samples from the human preference study, showing cases where the human judges preferred our score, NLS, or neither scores. In the latter case, the human judges preferred equal scores for Models A and B.

Dataset	Question	GT	Model A	Model B	Human pick
DocVQA	What is the vitamin A requirement (in I.U.) for a ’lactating’ mother ?	“1,000 i.u. plus basic requirements”	“basic requirements”	“1,000”	SMuDGE
			NLS: 0.54	NLS: 0.0	
			Ours: 0.0	Ours: 0.62	
MP-DocVQA	What is the day and date of Meeting?	“thursday 22 october”	“thursday”	“saturday 24 october”	SMuDGE
			NLS: 0.0	NLS: 0.74	
			Ours: 0.81	Ours: 0.25	
InfographicVQA	Which age group uses social media the most?	“18-29 year olds”	“18-29 group”	“18-24 year olds”	SMuDGE
			NLS: 0.53	NLS: 0.93	
			Ours: 0.98	Ours: 0.0	
DocVQA	What is the date of the letter?	“august 1, 1983”	“The date of the letter is August 1, 1983.”	“August 1983”	Neither
			NLS: 0.0	NLS: 0.78	
			Ours: 0.97	Ours: 0.0	
InfographicVQA	What is the estimated number (in billions) of social media users around the globe by 2019?	“2.72”	“#infographic”	“2. 72”	ANLS
			NLS: 0.0	NLS: 0.8	
			Ours: 0.0	Ours: 0.0	

Table 5.3: Top-5 models based on robustness rankings produced by ANLS versus our score (with $\alpha = 0.25$).

ANLS		SMuDGE	
1	Human	1	Human
2	QWen2-VL	2	QWen2-VL
3	InternVL2-Pro	3	InternVL2-Pro
4	QWenVL-Max	4	Molmo-72B
5	InternVL-1.5-Plus	5	Snowflake Arctic-TILT

5.4.5 Human evaluation

We used human judgment to assess the validity of our scores compared to ANLS. To do this, we used data from three benchmarks: DocVQA, MP-DocVQA, and InfographicVQA. In each benchmark, we sampled questions and a pair of answers produced by two models, indicated by model A and model B (different models could be selected for each sample). We limited the samples to cases where model A’s NLS score was higher than B, but SMuDGE scored B higher than A, or vice versa. We sampled up to 100[12] such question-answers triplets from each benchmark. Three researchers were presented with these triplets, as well as the ground truth answer, and asked which model they thought should be scored higher. The annotations produced a mean Cohen’s κ of 0.82, indicating a high level of agreement. We filtered the annotations to those on which at least two annotators agreed. This resulted in 28 samples for DocVQA, 86 samples for MP-DocVQA, and 66 samples for InfographicVQA.

Figure 5.7 shows the annotators’ agreement rates with NLS versus our score. The “Neither” bucket indicates that the annotators believed the models should have been scored equally. As the figure shows, annotators agreed with SMuDGE in the majority of cases across all three benchmarks, indicating that our approach is better aligned with human judgment. We observe that

[12] Some datasets had fewer qualifying triplets.

InfographicVQA, yielded has the highest rate of agreement with NLS, contains the largest number of misspelled numbers, as in the last row of Table 5.7, which could be a result of the complex layout and design of infographics.

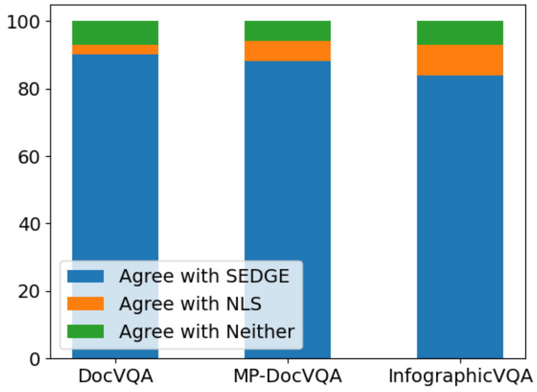


Figure 5.7: Human preference for pairwise rankings produced by NLS versus SMuDGE (with $\alpha = 0.25$).

5.5 Conclusion

In this study, we showed how popular evaluation metrics such as ANLS can miss important nuances when used to analyze Document VQA models. Instead, we proposed SMuDGE, a new metric that is sensitive to the groundedness of the models' outputs. Through extensive analyses, we showed how SMuDGE is better aligned with human judgement as well as the calibratedness of the models. Our analyses also showed that rankings produced by SMuDGE were better indicators of a model's robustness across question types and in different benchmarks. Our studies demonstrate the importance of groundedness in the performance and assessment of Document VQA models. We hope that in addition to presenting a new evaluation method, our study inspires researchers to develop better grounded Document VQA models.

Chapter 6

Conclusion and future work

In Section 1.1 we introduced a real-world scenario in which Alice, a knowledge worker at a financial firm is tasked with processing 1,000 authorized signatory forms. We examined the challenges that Alice would face if she were to use modern VRDU models to automate all or part of her process. Throughout this dissertation, we have introduced a series of studies that attempt to address one or more of the challenges faced by Alice. CompAQT and CounterComp, introduced in Chapter 3 enhance the compositional generalization of Quantitative Question Answering, hence enabling Alice to process questions that require multistep quantitative reasoning, such as calculating an authorized signatory's maximum authorization limit. APReCoT and AliGATr, introduced in Chapter 4, enable Alice to extract relevant information from the forms, guarantee localization, and provide well-calibrated outputs. SMuDGE, introduced in Chapter 5, allows Alice to evaluate Document VQA models more effectively, selecting those that provide answers that are better grounded semantically and multimodally.

In the remainder of this Chapter, we reflect on the advancements that the field of VRDU has made toward grounded document AI in the recent years, review the contributions of our research in detail, and lay out directions for future work.

6.1 Reflections on recent advancements in grounded VRDU

Since the conception of our work, the field of VRDU has evolved rapidly. Thanks to the diminishing cost of computation and new advancements in infrastructure design, data and memory efficiency are less central to achieving robust performance across tasks. In tandem, the popularity of Large Language Models and their multimodal counterparts has transformed the VRDU field. Adapting open-domain vision-language models to VRDU tasks has yielded models that have surpassed all prior SotA results [11, 24, 103].

As a result of these advancements, the lens through which we have explored efficiency via models such as APReCoT and AliGATr may no longer seem crucial to VRDU technologies or the broader field of multimodal AI. As an example, by extending the generative target of AliGATr from layout alone to layout *and* text, the model might achieve better performance at scale. Nevertheless, certain studies continue to challenge the assumption that efficiency is an increasingly nonessential topic, from the perspective of environmental impact [180], user privacy

[204], and on-device enablement [3, 181].

In the meantime groundedness continues to gain attention in the field, as a small but growing number of studies explore grounded multimodal reasoning for VRDU tasks. These models are inspired by prior work in grounded visual question-answering, which goes as far back as scene-graph representations [58]. Studies such as Point-and-Ask [112] and Connect Caption and Trace [117] successfully demonstrated that localizing answers within the input can improve performance on visual question answering. In the era of Multimodal LLMs, this idea was further developed to show improvements for reasoning over open-domain images [32] as well as user interfaces [101, 213].

More recently, studies such as LMDX [132] have begun to explore the task of localization for generative VRDU models. Localized VQA datasets, which were once limited to open-domain settings [135] are now being developed for VRDU tasks [203]. This slow but steady growth in the grounding-focused VRDU literature corroborates our argument for the importance of groundedness in the development of VRDU models, and their applicability to real-world settings.

6.2 Summary of our contributions

The core contributions of our work can be grouped into five main categories, each addressing one of the aspects of multimodal reasoning that are key to any enterprise VRDU model:

1. **Grounding the model's attention patterns in quantitative language (Section 3.1)**: We showed that self-supervised attention units can learn spurious patterns that undermine compositional generalization when performing multi-step question answering over quantitative data. Instead, we proposed *CompAQT*, an attention mechanism that is explicitly grounded in quantitative expressions, and showed that it can improve reasoning over multi-step operations. Further, we contributed a unified collection of quantitative QA benchmarks based on four popular datasets with diverse schemata.
2. **Grounding the model's quantitative reasoning by counterfactually augmenting the search space (Section 3.2)**: We extended CompAQT's reasoning performance by augmenting the search space using counterfactual sampling. This not only further enhanced the model's compositional generalization, but also improved its performance on OOD samples. The resulting model, *CounterComp*, can be used in small-data regimes, or when training samples are skewed towards few-step operations, both of which are scenarios that are common in enterprise settings.
3. **Grounding the model's visual reasoning in the document's design (Section 4.1)**: We demonstrated that complex visual backbones inspired by open-domain vision-language models were inefficient and unnecessary for VRDU tasks. Instead, by factorizing the visual signal into clusters, we were able to recognize the color palette of visually rich forms. We proposed *APReCoT*, a small and efficient model that is able to learn contrastive representations and performs on par with SotA models on processing forms.
4. **Grounding the model's spatial reasoning in the document's layout (Section 4.2)**: We demonstrated that graphs provide a rich structure for modeling complex layouts in documents that follow grid-based structures (such as forms). Based on this finding, we de-

veloped *AliGATr*, a parameter and data-efficient graph generation model that can perform exrtactive tasks (KIE) as well as associative tasks (RE) on visually complex forms, and can localize its outputs. AliGATr can be used as a robust, lightweight, and well-grounded alternative to SotA models. Because AliGATr uses graphs as abstractions, it is not bound to certain characteristics of the input image such as its resolution and dimensions, which is a challenge that continues to impact SotA vision-language models [90].

5. **Enabling grounding-aware evaluation methodologies (Chapter 5)**: Lastly, we brought together our previous work on spatio-visual and quantitative reasoning by proposing *SMuDGE*, an evaluation methodology that accounts for groundedness in all three aspects. By allowing flexibility in how SMuDGE is calculated, enterprise practitioners can adapt the metric to the requirements of their downstream applications.

Reasoning	**Grounding the design**	**Grounding the objective**	**Grounding the search space**
Quantitative		Unsupervised alignments (Section 3.1) Metric learning (Section 3.2)	Counterfactual sampling (Section 3.2)
Spatio-visual	Graph-based generation (Section 4.2)	Cluster membership (Section 4.1)	
All	Grounded evaluation (Chapter 5)		

Table 6.1: An updated view of Table 2.3, where our contributions have been added to corresponding cells, blue. Each highlight has a reference to the section where it is covered.

As suggested in Chapter 1, multimodal grounding can be encouraged by modifying the design of VRDU models, by defining grounding-aware learning objectives, or by scaffolding the search space. Table 6.1 provides an overview of the contributions that we have made to the field of grounded enterprise VRDU from the perspective of these intervention methodologies. Throughout this thesis, we have explored all three aspects of intervention, covering venues that were previously under-explored. Most notably, in the category of grounding-by-design, while graph-based structures have been used in the past to capture spatial and visual signal (as evidenced by Table 2.2), generative approaches to graph-based representation have not been examined. The resulting graph-encoder models suffer from poor calibration, a problem which we addressed by proposing *AliGATr*, an auto-regressive graph generator.

Additionally, studies that modify the learning objectives of VRDU models to encourage multimodal grounding have largely followed reconstructive objectives akin to Masked Language Modeling, which sample random segments of the input. In contrast, we have proposed a suite of new methodologies that target specific components of the input, with the goal of improved compositional generalization, robust OOD performance, and better calibration.

Lastly, augmenting the search space of VRDU models as a way to improve their grounding within the input has been an under-explored venue. By proposing a new methodology to augment the search space using counterfactually sampled data, we open the door to new self-supervised techniques in small data regimes, especially when compositionality largely governs the distribution of samples.

All of the above contributions are crucial to the adaptation of VRDU models in enterprise settings. We hope that the thesis' contributions to this domain encourage further research into grounded multimodal reasoning for enterprise documents.

6.3 Limitations of our work

As demonstrated throughout our work, groundedness in multimodal enterprise document understanding can take on many forms. In the spatio-visual modality, groundedness can be interpreted as localization of information. Localization is most relevant to extractive tasks such as Information Extraction or Extractive VQA, where every piece of information in the output needs to be traceable to the input. Extending this idea to abstractive tasks such as Classification or Abstractive QA requires mapping out in detail the reasoning steps required to address the task. For example, if certain segments of a document are most relevant to a question, a well-grounded model should be able to identify those segments as such. This is a complex and semantically underdefined challenge that has been tackled in the unimodal domain [69] as well as multimodal QA outside of VRDU [58]. We consider this task to be outside the scope of our research, though we hope that our work has provided sufficient motivation for VRDU researchers to examine it.

In enterprise VRDU, semantic groundedness can go beyond multimodal localization. As we observed in Chapter 5, models that are not grounded in semantic categories such as numbers can produce errors that are just as undesirable as hallucinations. In examining semantic groundedness, our work in Chapter 5 did not examine semantic categories beyond textual/numeric/hybrid forms, such as currencies, dates, timestamps, etc. each of which come with nuances that can have impacts on downstream applications if mishandled or mis-produced. Complex constructs such as tables further complicate the evaluation of VRDU models. If a model is expected to produce a table or a snippet of a table as a response, a robust evaluation metric should be invariant to the ordering and rendering of the content within the table. We leave the exploration of such complexicites to future work.

Additionally, most of our work on document-grounded VQA is focused entirely on single-span, extractive answers. To extend the grounding mechanism to multi-span answers, our algorithms would need to handle an arbitrary number of possible answer spans within the document, which we leave to future work.

In the context of quantitative reasoning, despite impressive advancements in SotA performance, research in certain domains such as Financial QA remains nascent. As such, benchmarks such as FinQA and TAT-QA only scratch the surface of the challenge of multi-step quantitative reasoning. Our proposed QA models in Sections 3.1 and 3.2 are not intended for settings where complex or high-level quantitative insights are required (e.g. anomaly or trend detection). In such settings the large space of operations makes it challenging to rely exclusively on soft alignments between the input and output. This is analogous to the challenge of localization in VQA, where the more “abstractive” a question, the more challenging it is to ground the reasoning within specific segments of the document. The multi-faceted nature of this challenge calls for new research into natively grounded multimodal reasoning over enterprise data.

Additionally, our work does not fully explore the extent to which compositional modeling of language can improve quantitative reasoning in multi-step Financial QA. Since CompAQT and CounterComp use soft alignments to enhance compositional generalization, they merely approximate the decomposition of arithmetic components using simple heuristics. While other distant-learning methods continue to dominate the literature on mathematical reasoning [41, 140, 177], further work in this domain may build on alternative approaches such as neural-symbolic architectures [62, 100] or preference-optimization methods [160].

In addition to limitations specific to our work, the field of VRDU is constrained by broader limitations within the literature. The following sections provide an overview of these limitations in the context of four key components of research in enterprise document understanding, namely, data, model, and evaluation methods.

6.3.1 Data limitations

Image documents, especially in the enterprise domain, are often owned by specific legal entities and are therefore not always available for redistribution, even when they do not include confidential data. This makes open-domain collections of public documents difficult to source compared to unimodal corpora such as Wikipedia, Common Crawl, and social media posts. As a result, research in the domain of multimodal document understanding is dominated by a limited set of corpora, as demonstrated in Table 6.2.

Unsurprisingly, due to licensing restrictions, the US government is the leading supplier of VRDU datasets.[1] Regulatory disclosure protocols and legal settlements have resulted in collections such as IIT-CDIP [91], Kleister [166], and DeepForm [170], which cover documents from regulated industries such as tobacco manufacturing and political advertising. This poses two challenges to our research. First, these public collections are not always representative of the variety of enterprise documents, even in English-language jurisdictions. As an example, authorized signatory forms are often not represented in any of the datasets due to their confidentiality.

Second, many datasets used in financial quantitative reasoning studies including those used in our research were created based on financial reports, with many focusing specifically on regulatory disclosures provided by the United States Securities and Exchange Commission. As such, these data assets are biased towards attributes and patterns expected in such reports, including GAAP metrics[2], currency units, and left-to-right orientation for tabular structures.

Additionally, VRDU datasets that cover the KIE task are often evaluated using a standard IOB schema. Originally developed for the IE task in unimodal text, the IOB schema honors a sequence order that is inherent to unimodal text. In contrast, multimodal documents are 2-dimensional artifacts, and a canonical ordering of the words might not be readily available due to their complex structure.

Nevertheless, in order to support the IOB schema, many VRDU datasets provide a proposed ordering as part of their annotations, which the models in turn use during evaluation. This leads to a fundamental problem of information leakage—the datasets are providing information to the model regarding the order of the words which would not be available in real-world test settings.

Lastly, as pointed out by Lee et al. [89], some models assume that they have access to segment-level bounding boxes at test time. They demonstrated how the performance of LayoutLMv3 on the FUNSD dataset would decrease by almost 10 points when access to segment-level information wasn't provided (see second row of Table 6.4).

Recent studies such as Zhang et al. [217] have taken steps towards addressing this problem by proposing graph-based models that are sensitive to reading order. The authors have also released revised versions of the FUNSD and CORD datasets to address the problem of information

[1] See Figure 6.1 for an illustration of the lineage of the datasets listed in Table 6.2.

[2] https://www.cfainstitute.org/en/advocacy/issues/gaap

Dataset	Citation	Training size	License	Upstream publisher	VrDU tasks
IIT-CDIP	[91]	6,910,192 docs	Fair Use	US Gov.	None
RVL-CDIP	[45]	400,000 pages	Fair Use	US Gov.	CLS
DocLayNey	[133]	80,863 pages	CDLA-Permissive	Unknown/varied	SEG
DocILE	[162]	106,680 docs 108,715 pages	Fair Use	US Gov.	KIE LIR
DocVQA	[113]	12,767 pages	Fair Use	US Gov.	VQA
DUDE	[179]	5,019 docs	Unspecified	Unknown/varied	VQA
BuDDIE	[233]	1,665 pages	Proprietary	US State Govs.	KIE
FUNSD	[63]	199 pages	Custom	US Gov.	KIE RE
CORD	[128]	2,000 pages	CC-BY-4.0	Businesses	KIE
SROIE	[57]	1,000 pages	CCA 4.0	Businesses	KIE
DeepForm	[170]	∼20,000 docs	MIT	US Gov.	KIE
Kleister	[166]	3,318 docs 64,872 pages	OGL	US & UK Govs.	KIE
VRDU	[195]	2,556 docs	Fair Use	US Gov.	KIE
Payment	[89]	∼10,000 docs	Proprietary	Google	KIE

Table 6.2: 14 popular datasets in the VrDU literature, used by the models in Table 6.3. VrDU task legend: `CLS`: Document classification. `SEG`: Page segmentation. `KIE`: Key information extraction. `VQA`: Visual question answering. `LIR`: Line item recognition. Note: The table excludes `OCR` datasets as well as those focused on historical document understanding.

leakage. Further investigation in this domain will enhance real-world outcomes for downstream users.

6.3.2 Model limitations

SotA models inherit the licensing challenges of the datasets that they have been trained on. Additionally, the models carry their own Intellectual Property, which might restrict their use. Moreover, the unavailability of open-sourced code and/or model weights further limits usage in downstream tasks. As Table 6.3 shows, only 4 of the 7 SotA models on the list have any open-sourced components, and no model has all three components that are required for it to be considered fully open-source (i.e. pre-training code, pre-trained weights, fine-tuning code).

This has limited our ability to recreate baselines in several studies presented throughout this thesis, and has led us to rely heavily on self-reported performance measures in other publications.

6.3.3 Evaluation limitations

The benchmarks in the VRDU field often calculate a model's performance as an average over all fields within the dataset, whereas in real world settings, performance is often measured at

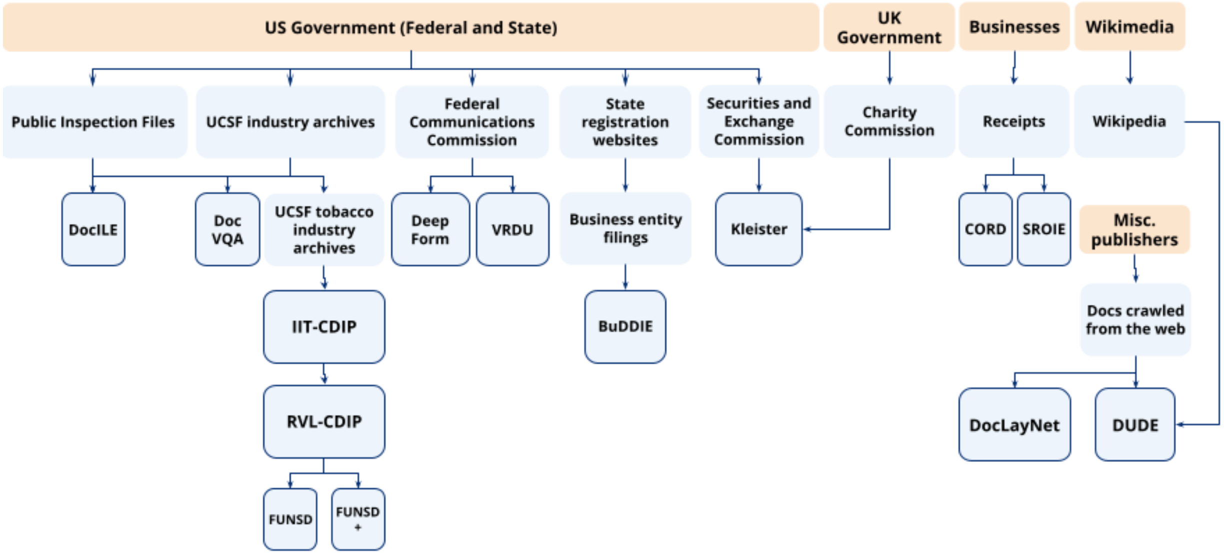


Figure 6.1: The lineage of the datasets listed in Table 6.2. Each dataset is displayed in a bordered box. The remaining boxes represent upstream sources of documents, with the most upstream publisher highlighted in orange.

the document level. As an example, Table 6.4 shows the performance of LayoutLMv3 when measured overall, compared to context-specific measurements such as doc-level accuracy and average F1 per document. Doc-level accuracy shows the percentage of documents that can be processed in a “straight-through” fashion, i.e. ones which the model processes without any errors. As the table shows, only 4% of documents are processed by the model without any errors. On the other hand, LayoutLMv3’s average F1 per document is 83.08. This means that any human reviewers that remain in the loop can focus their effort on reviewing the portions of each document that the model is likely to mishandle. They can strategize by analyzing the model’s performance per entity type, and focus their efforts on the “Header” category for which the model performs poorly compared to other entity types. Alternatively, given a model that produces well-calibrated probabilities, they can focus their efforts on low-confidence outputs.

Throughout this thesis, in order to remain consistent with the literature and establish consistent benchmarks, we have reused standard metrics (`Precision`, `Recall`, `F1`, etc.) to report on the performance of our proposed models. However, in order to accommodate many downstream applications in the enterprise domain, a more nuanced approach might be required, as argued above.

6.4 Future directions

Given the challenges laid out in the previous section, we will now discuss the opportunities that they offer to researchers and practitioners in the field. We will present these opportunities in the context of several high-level focus areas, each covering one or more of the research challenges discussed in the previous section.

Model	Citation	# Params	Architecture	License	Commercial Affiliate	OSS Status	Generative/ Grounded	VrDU tasks
LayoutLMv3$_{\text{LARGE}}$	[56]	368M	TR	CC BY-NC-SA 4.0	Microsoft	PW FW FC	N/Y	CLS SEG KIE VQA
UDOP	[171]	794M	TR	MIT	Microsoft	PW FC	Y/N	CLS KIE TR VQA
FormNetV2	[89]	204M	GR	N/A	Google	None	N/Y	KIE
UReader	[209]	86M*	MLLM	Apache 2.0	Alibaba	PW PC	Y/N	KIE TR VQA
DocLLM	[189]	1B, 7B	MLLM	N/A	JP Morgan	None	Y/N	CLS KIE TR VQA
Qwen-VL-MAX	[11]	Unknown	MLLM	Custom	Alibaba	FC	Y/N	VQA
SMoLA-PALI-X	[199]	Unknown	MLLM	N/A	Google	None	Y/N	VQA

Table 6.3: Models with SotA performance on a variety of VrDU tasks as of Jan, 2024. Architecture legend: TR: Transormer-based. GR: Graph-based. MLLM: Multi-modal LLM. OSS Status legend: PC: Pre-training code. PW: Pre-trained weights. FC: Fine-tuning code. FW: Fine-tuning weights. VrDU task legend: CLS: Document classification, SEG: Page segmentation, KIE: Key information extraction, TR: Tabular reasoning, VQA: Visual question answering. *UReader reports its number of trainable parameters, but the model is created by applying LoRA [53] to mPLUG-Owl [210], which has around 7B parameters.

6.4.1 Data Curation

The inherent issue of copyright and ownership that limits the use of document collections in training VRDU models is compounded by the confidentiality of content in most enterprise settings. Recent work that has focused on synthetic document generation explores the possibility of creating realistic layouts [146], content [48], or both [10]. A challenge in using synthetic documents for VRDU tasks is that many of them are modeled after the same public-domain datasets mentioned in Table 6.2, and are therefore likely to carry the same biases in their multimodal signal. To tackle this problem, a two-pronged approach is needed: 1) Enterprise researchers and practitioners can take on a leading role in releasing synthesized collections that reflect their proprietary documents with high fidelity, without violating their confidentiality. 2) Public-domain collections such as the IIT-CDIP dataset can be augmented with a larger variety of enterprise collections from a wider range of industries and time spans. As Figure 6.1 illustrates, the datasets that are derived from a variety of upstream sources are rare, and curators often focus on one or two specific publishers.

Additionally, datasets can be curated with better localization signal. When annotating datasets, we encourage researchers to use tools that support visual annotations, such as PAWLS [121]. Developing models that provide bounding-boxes as part of their output further enables downstream users to verify them.

Reported F1 (with segments)		92.08
True F1 (no segment info)		82.86*
F1 per entity type	**Header**	57.49
	Question	86.03
	Answer	83.25
Doc-level accuracy		4%
Avg F1 per doc		83.08

Table 6.4: Evaluation of LayoutLMv3 performance as reported in Huang et al. [56] (Reported F1) versus when using contextualized metrics (e.g. Doc-level F1). *The "True F1" value of 82.86 is consistent with the value reported in Lee et al. [89], i.e. 82.53.

6.4.2 Model Calibration

We encourage the development of new benchmarks that probe a model's output or its internal representations from a calibration perspective. In the unimodal literature, previous studies have often done so by presenting evidence that the model generates compositional representations [38], that the model's confidence aligns with its performance [66], or that the model makes "forgivable" errors [152]. Similar measures can be used by VRDU researchers to demonstrate whether the models produce well-calibrated outputs. Below are some research questions that can probe the issue of calibration in the VRDU domain:

- Is the model able to map documents to a compositional semantic space, where similar documents (in terms of content) are grouped together? How about similar documents in terms of visual style? In terms of layout? In terms of category (e.g. forms versus contracts)? Or in terms of issuer/producer?
- Does the model produce well calibrated probability distributions as part of its output? If not, does it lend itself to a post-hoc calibration approach such as in Jiang et al. [66]?
- Does the model's performance linearly scale with its confidence?
- Can the model's errors be identified at test time using contextual signals? For example, does the model consistently do well on tables but poorly on diagrams?
- How can the model be integrated into an operational pipeline where human oversight can be directed toward high-risk samples?

6.4.3 Contextualized Performance

We encourage researchers to report performance metrics not only at the dataset-level, but also at the document (and possibly output-type) levels. This can be used as an estimate of the amount of time that a knowledge worker can save by using the model in an operational pipeline, provided that the model is well calibrated.

Performance can also be profiled based on the category, type, visual complexity, and contents of the input. As an example, the DocVQA benchmark provides a breakdown of the different classes of questions and the type of reasoning that is required to answer them (e.g. reasoning over tables, charts, layout, or text). This can facilitate a more purposeful analysis of each model's performance compared to a singular measurement.

6.4.4 Building on recent advancements

Lastly, we reflect on the progress that neural architectures and large-scale training paradigms have made in recent years, as described in Section 6.1. The success of Large Language Models in tackling foundational tasks in natural language processing has led researchers to explore their utility in enterprise applications [33, 81, 150, 155, 196, 201, 215, 225]. In settings that require domain expertise or multifaceted analyses, these models have been reimagined as autonomous or semiautonomous agents that can carry out a complex task by tackling a single component at a time [202]. These agents are often coordinated through a Mixture-of-Experts model [21] or a collaborative agentic framework [97]. Due to the complexity of the input and the task, Financial QA lends itself to this approach, and recent studies have explored agent-based frameworks with some success [39, 43]. Nevertheless, SotA performance quickly degrades in contexts where deep domain expertise is required [111], or where compositional generalization plays a major role [134, 165].

Meanwhile, in adjacent domains that rely on symbolic reasoning such as math-word problem solving or code generation, researchers have made breakthroughs by borrowing methods from the Reinforcement Learning literature to tune LLMs. The success of preference optimization methods such as PPO [1], DPO [143] and GRPO [160] has established reward modeling as a promising direction in augmenting LLMs' symbolic reasoning capabilities. This offers a major opportunity to researchers in the Financial QA domain to explore similar approaches. Early studies have shown promise [138], but impact continues to be constrained by the limited availability of domain-specific data [108].

More recently, Qwen et al. [140] have demonstrated that combining a reward model with a data synthesization technique can lead to robust performance in mathematical reasoning. This is consistent with our findings in 3.2, where we showed that careful augmentation of the model's search space through counterfactually sampled data can enhance its robustness in Financial QA. Combining CounterComp's sampling strategy with preference optimization methods might further enhance the model's performance in challenging contexts.

In tandem, research in multimodal groundedness has continued to gain traction in the literature [32, 132, 213] and some researchers have explored the adaptation of agentic frameworks to the navigation of user interfaces [80, 101, 186] as well as enterprise documents [159, 215, 223]. Layout-aware approaches, such as AliGATr, offer an advantage in that they can be adapted to accommodate visually rich web pages as well as documents, allowing enterprise users to seamlessly navigate the multitude of information resources, from confidential financial disclosures that are available in document form, to publicly available disclosures that are posted to regulatory websites and portals.

Agentic frameworks can not only move the field in the direction of higher performance, but they may also create new opportunities in effective Human-Machine collaboration in enterprise settings, allowing humans to optimize their workflows toward tasks that require deep contextual expertise. The maker-checker protocol that governs many document-grounded applications in enterprise settings [18] lends itself to a productive model of Human-Machine collaboration, where the LLM agent can act as a maker, and the human expert, acting as the checker, provides feedback to tune the LLM on an ongoing basis. To do this, the field needs to prioritize the pursuit of grounded methodologies that produce localized, contextualized, and well-calibrated outputs.

We began this thesis by presenting a scenario in which Alice, a knowledge worker at a financial firm, is tasked with processing a collection of enterprise documents. We demonstrated that semantic and multimodal groundedness were key aspects of any automation tool, without which Alice's workflow would continue to be burdened by laborious data extraction, normalization, and entry tasks. By building on current advancements, the field of VRDU can move in the direction of producing methodologies that allow Alice access to tools that provide her with the desired output, where information is processed holistically, outputs are localized, errors are highlighted, and her workflow is elevated to that of an expert reviewer. We hope that our research on grounded multimodal enterprise document understanding has offered a step in that direction.

Disclaimer

This work was supported by JPMorgan Chase & Co. The manuscript was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates (“JP Morgan”), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

Appendix A

Chapter 3 Appendices

A.1 Dataset unification

In order to unify the datasets, we convert each of them into the same format. We choose FinQA as the reference format, since it encodes multi-step programs in a standardized representation. Each program is encoded as a right-expanding binary tree where each operation is guaranteed to have two operands, with one of more operands set to `NONE` if necessary.

FinQA provides two versions of each data-table: one with the raw format, and one where the content has been normalized such that all row headers are merged into one row header, and all column headers are merged into one column header. This removes discrepancies in the way tabular data is represented across different samples. The dataset also includes gold facts from each table and surrounding text. The facts have been tokenized and verbalized such that the model can easily map operands to number expressed in them.

Transforming other datasets to match the FinQA format requires following three steps: 1) Normalizing the tables, 2) Normalizing the programs, and 3) Normalizing the evidence. Below, we describe these steps in detail.

A.1.1 Normalizing the tables

In the TAT-QA and HiTab, each table is represented as a nested array. In MULTIHIERTT, each table is represented by the raw html code, which is easily convertible into a nested array.

The top n rows and the left m columns of each table form the column and row headers, respectively. To find n, we follow Algorithm 1. A similar method is applied to determine m. We then merge the contents of the top n rows and the left m columns to form a singular column header and a singular row header.

A.1.2 Normalizing the programs

MULTIHIERTT represents programs in a format that is compatible with FinQA (e.g. `subtract(20, 30), divide(#0, 20)`). TAT-QA and HiTab represent them in a non-standardized format (e.g.

Algorithm 1 Column header finder

```
$N \leftarrow$ num_rows
$K \leftarrow$ num_cells_in_first_row
non_empty_cells $\leftarrow$ []
for $i \in \{1, \dots, K\}$ do
    if table$[1][i] \neq$ empty then
        non_empty_cells$[i] =$ TRUE
    else
        non_empty_cells$[i] =$ FALSE
    end if
end for
$n \leftarrow 2$
while $n \leq N$ and FALSE $\in$ non_empty_cells do
    for $i \in \{1, \dots, K\}$ do
        if table$[n][i] \neq$ empty then
            non_empty_cells$[i] =$ TRUE
        else
            non_empty_cells$[i] =$ FALSE
        end if
    end for
    $n := n + 1$
end while

return$n$
```

`(20-30)/20` or `1-30/20`). We use an Abstract Syntax Parser[1] to process each expression into a tree. We then programmatically traverse the tree to generate a FinQA-compatible program. Next, we match each operand to the evidence. If not found within the evidence, an operand is replaced by a constant (e.g. `multiply(#0, 100)` → `multiply(#0, const_100)`.

A.1.3 Normalizing the evidence

We use a tokenizer similar to the FinQA tokenizer to process each sentence and table within each passage. This is to ensure that the operands are guaranteed to match the numbers mentioned in the evidence.

Table A.1 shows additional filters applied to each dataset.

Dataset	**Configuration**
FinQA	no filters
TAT-QA	arithmetic category only
HiTab	no multi-span answers at least one operation required
MULTIHIERTT	arithmetic category only no span-based answers

Table A.1: How each dataset was filtered to include in the unified collection.

A.2 Pointer Verbalizer Network

The programs are composed of two types of tokens: 1) operator tokens, which are sampled from a symbolic space (i.e. math symbols), and 2) operators, which are either numbers mentioned in the facts, or constants such as 100 or 1,000,000 for scaling the output. Models such as FinQANet treat the program as a sequence of tokens, not differentiating between operators and operands. At each step a new token is sampled from the universe of all possible operators and operands, and a mask is used to make sure two operands are generated after each operator. The process stops once the `EOF` operator is generated.

In contrast to FinQANet, our Pointer Verbalizer Network uses a two-pronged approach that accounts for the differences between the operators and operands, and employs verbalization to enhance performance.

A.2.1 Generating operators

The generator can be regarded as an encoder-decoder model akin to sequence-to-sequence models employed in Neural Machine Translation (NMT). However as opposed to NMT, the operators are sampled from a symbolic space (i.e. math symbols), which does not have the same distribution or compositionality as the input space (i.e. text). A similar problem exists in most

[1] We use the standard python3.7 ast module. `https://docs.python.org/3.7/library/ast.html`

semantic parsing tasks, but it is sometimes alleviated by mapping the target domain into a space that is close to the input domain. For example in a Text-to-SQL task, instead of referring to “table_#3.column_#1”, the names or descriptions of the table and column are used (e.g. “students.name”). This allows the model to leverage compositional semantics in the target domain as well as the source domain.

We pursue a similar strategy by mapping the program operators into text, i.e. verbalizing them. For example, instead of the categorical symbol `divide`, the token “divide” is used to represent the division operation. As a result, the operator generator produces a sequence of tokens. These tokens can be mapped to the nearest operator based on their cosine similarity. The loss is thus calculated as:

$$\mathcal{L}_{\text{operator}} = \gamma \mathcal{L}_{\text{CE}} + (1-\gamma)\mathcal{L}_{\text{reg}} \tag{A.1}$$

where γ is a hyperparameter, $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss between the predicted operators and the true operators, and $\mathcal{L}_{\text{reg}}$ is a regression loss defined as the sum of cosine distance and MSE loss between predicted operator token embeddings $\mathbf{o}_i$ and the true operator token embeddings $\mathbf{a}_i$.

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N} \text{cosine}(\mathbf{o}_i, \mathbf{a}_i) + \text{MSE}(\mathbf{o}_i, \mathbf{a}_i) \tag{A.2}$$

The regression loss functions as a regularizer to ensure that the verbalized predictions do not stray too far from true operator tokens. In our experiments, we set $\gamma = 0.8$.

A.2.2 Generating operands

As opposed to the operators, the operands are always selected from a list of existing numbers or constants. This, along with the verbalization of operators allows us to approach operand-generation using a pointer network [187]. At each step, the predicted operator token embedding $\mathbf{o}_i$ is used as the hidden state, and the model selects the top two options from the list of possible operands.

Figure A.1 illustrates the proposed architecture. The top part of the figure shows how the sequence of operators id generated. The cross-attention mechanism helps augment the question with information from the facts. The output is then attended to by the verbalized vocabulary of operators. The output of this step is used in a recurrent network to predict the operator in each step.

The bottom part of the figure shows how the operands are selected for each predicted operator. This time, attention is used to augment the fact representation with information from the question. Using this, as well as the output of the operator predictor, the model uses a pointer network to select operands from a list of possible numbers and constants.

A.3 CompAQT’s experimental details

Tables A.2 lists the settings and parameters for each baseline. The settings used for FinQANet and TAGOP are based on [26] and [227], respectively. For all experiments that involved CompAQT, α and λ were both set to 0.1.

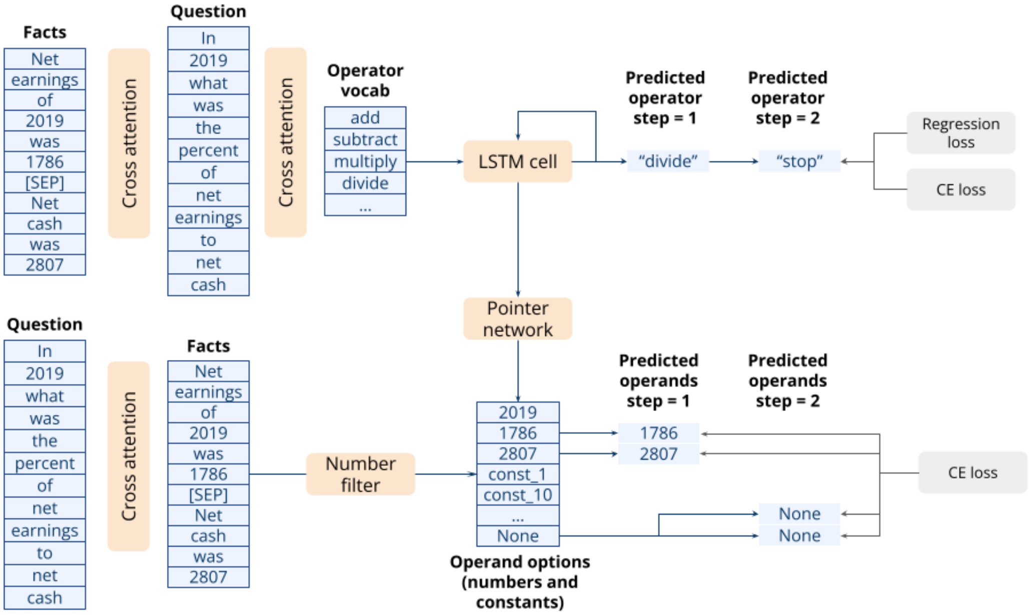


Figure A.1: The architecture of the Pointer Verbalizer Network. The top half shows the operator predictor and the bottom half shows the operand predictor.

All experiments were conducted on a machine with 8 NVIDIA T4 GPUs with 16GBs of memory per GPU.

Parameter	**FinQANet**	**TagOp**	**PVN**
encoder	RoBERTa-large[107]		
batch size	16	32	64
learning rate	2e−5	5e−5	2e−5
optimizer	Adam [77] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$		
epochs	100	50	50

Table A.2: Settings used for FinQANet experiments.

A.4 CompAQT experiments using retrieved facts

Table A.3 shows ablation results similar to Table 3.7 but for retrieved facts (instead of gold facts). Similar trends hold when using retrieved facts, with CompAQT modifications having a larger impact on multi-step programs.

Model	Program accuracy			
	1 step	2 steps	3+ steps	Overall
FinQANet	64.13	57.03	20.56	58.30
+self-attention	+3.01	+1.97	+1.95	+2.44
+alignment loss	+0.01	+4.66	+1.83	+2.00
+coverage term	+0.00	+2.02	+0.55	+0.31
+linear decay	+0.00	+0.10	+.48	+0.00

Table A.3: Ablation results on the FinQANet model, applied to the FinQA dataset with retrieved facts.

A.5 Operation-aware pre-training

The FinQANet generator performs better on examples with simpler programs which contain 1 or 2 steps compared complex programs with 3 or more steps. We perform masked language modeling (MLM) finetuning on the FinQANet language encoder over FinQA and TAT-QA data to see how it can encourage the generator perform better on complicated programs. First, we take the dev sets of FinQA and TAT-QA, and split both dataset 80/20 for training and testing masked token prediction. The resulting train and test set sizes are 1136 and 284, respectively. Second, we finetune Roberta [107] using three masking methods.

- **op**: mask one operator in every program
- **const**: mask one constant/operand in every program
- **op+const**: mask either an operator (50%) or a constant/operand (50%) in every program

For each example in our new dataset, we encode the natural language question, supporting facts, and program in a sequence. For each experiment, we finetune Roberta until it achieves 70-80% accuracy in guessing the masked tokens. Finally, we train the FinQANet program generator with our finetuned Roberta model. During inference we use the finetuned Roberta encoder in the FinQANet generator.

Table A.4 shows the experiment results. The baseline program out-performs most of the finetuned program generators except on the FinQA test set. In all experiments, masking the program operators (e.g., add, subtract, divide) leads to better performance compared to masking program operands or masking both operators and operands. A possible explanation may be that there are only 10 program operators compared to many possible operands. Furthermore, given the small size of our data used for finetuning, masking different types of tokens with a large range of values may confuse the finetuned model, and impair its ability to learn ties between text semantics and the desired output program. This intuition is reflected in figure A.2 which shows the loss and accuracy curves of the op, const, and op+const experiments compared to the baseline program generator.

We conduct further analysis on the generated programs with respect to program complexity to better understand how finetuning the Roberta encoder affects the generated programs. As shown in figure A.3, the baseline program generator performs better than most finetuned program generators in program accuracy in 1-step and 2-step programs. However, the finetuned program

Finetune	FQA-Valid	FQA-Test	TQA-Valid
baseline	62.29	61.90	71.32
op	61.61	**62.34**	70.35
const	60.02	58.24	71.32
op+const	58.55	57.9	68.60

Table A.4: Program accuracy of FinQANet program generator with baseline and finetuned Roberta encoders on the FinQA dev set. FQA-dev: results on the FinQA dev set at 25 epochs. FQA-test: results on the FinQA test set at 300 epochs. TQA-dev: results on the TAT-QA dev set at 50 epochs.

generators perform better than the baseline program generator in program accuracy on $\geq$ 4-step programs on both the FinQA dev and test sets. A possible explanation is that while the baseline program generator is competitive in overall program accuracy, it is biased to performing well on shorter programs due to the distribution of program lengths in the FinQA dataset. Thus, finetuning the program generator can help the model generalize better to complex programs at a slight cost in performance on shorter programs (as indicated by performance of $\geq$ 4-step programs).

A.6 CounterComp results on retrieved facts

Table A.5 shows the performance of FinQANet versus FinQANet+CounterComp on retrieved facts from the FinQA dataset. Similar to gold facts, CounterComp improves program accuracy, especially on multi-step output.

Model	**Program accuracy**			
	1 step	**2 steps**	**3+ steps**	**Overall**
FinQANet	64.13	57.03	20.56	58.30
+CounterComp	**67.52**	**58.87**	**22.79**	**61.18**

Table A.5: Ablation results on the FinQANet model, applied to the FinQA dataset with retrieved facts.

A.7 CounterComp training algorithm

Algorithm 2 details our pre-indexing, sampling, and training processes. Note that the algorithm is a simplified version of our implementation, e.g. it follows a basic SGD instead of a batch SGD process, and shows the process for only one epoch.

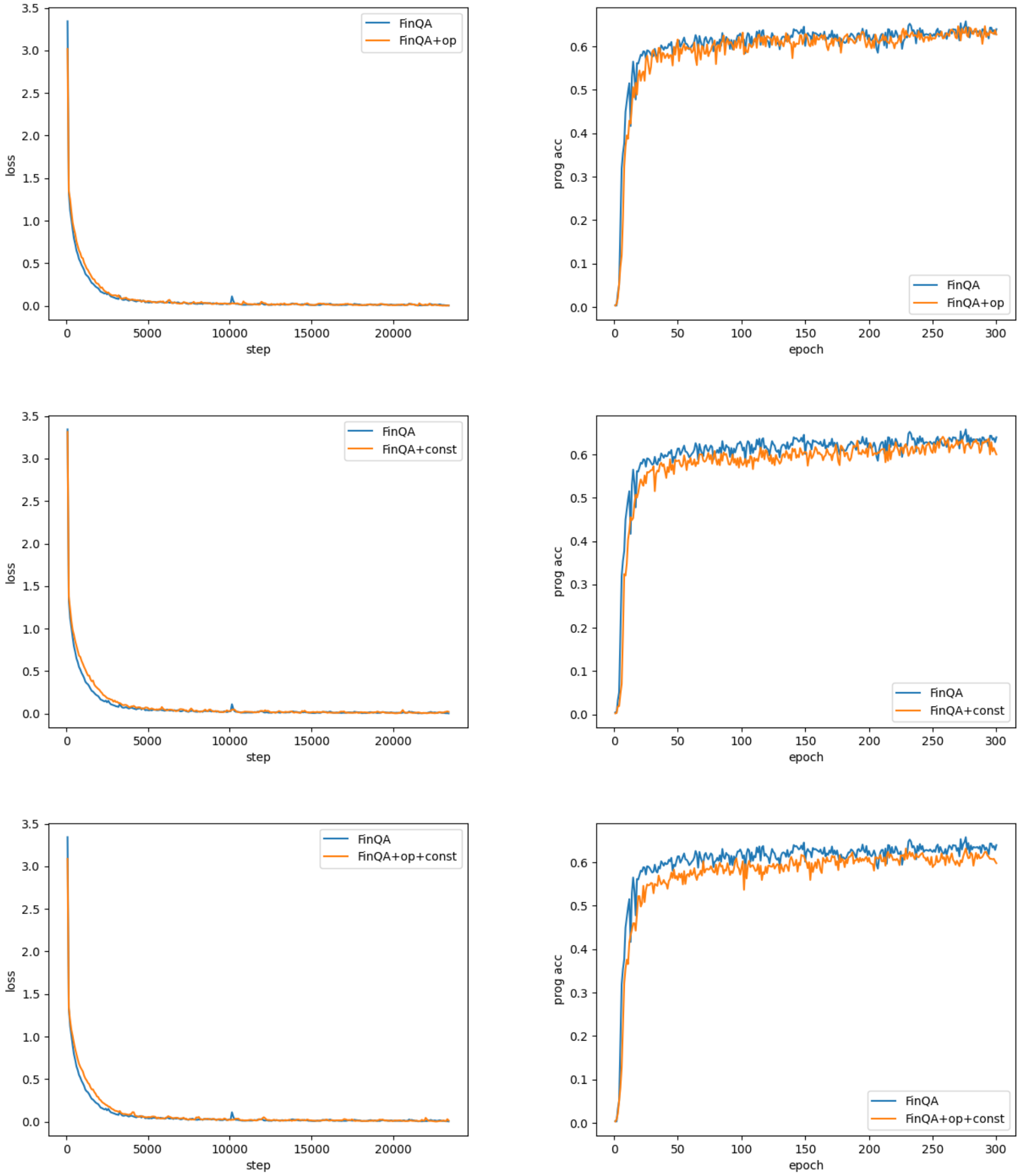


Figure A.2: Loss and accuracy curves of finetuned program generators on FinQA on the validation set

A.8 CounterComp for operators versus operands

CounterComp intervenes on operators, whereas operands provide another possible intervention target. As mentioned in Section 3.2.1, Learning to Image (L2I) [93], which focuses on counterfactual scenarios for operands, was able to outperform TAGOP by a large margin. L2I was evaluated on TAT-QA, a dataset with a limited set of possible multi-step operations, resulting in the challenge of compositional generalization being mainly focused on operands. Since we

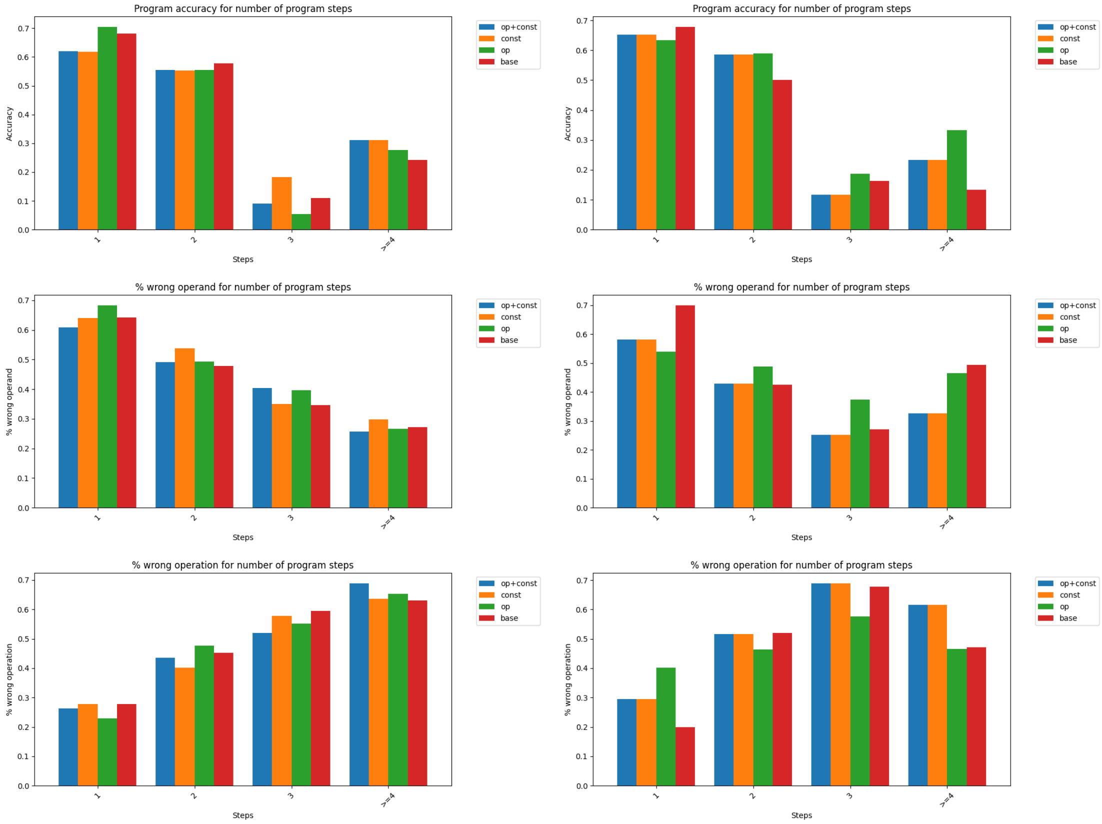


Figure A.3: Program accuracy, % wrong operands, and % wrong operations with respect to number of steps of finetuned program generators on the FinQA dev set (right column) and test (left column).

were not able to recreate the results reported in the original L2I paper[2], instead we evaluate an operand-focused approach via a metric learning method similar to CounterComp.

Given an anchor, we generate new samples using the operations laid out in the L2I paper (i.e. SWAP, ADD, MINUS, etc.), where one or more operands are perturbed in random. We apply the same perturbation in the facts. This effectively eliminates the ”imagination” component but provides a baseline that is more comparable to CounterComp. These samples are used as positive examples, whereas negative examples are randomly sampled from the batch.

Table A.6 shows the program accuracy of CounterComp versus the new method when applied to each dataset. As expected, TAT-QA is the only dataset responsive to the perturbation of operands. All other datasets suffer from an exclusive focus on operands. For HiTab and MULTIHIERTT, the operand strategy also underperforms compared to the baseline FinQANet performance (see Table 3.13).

[2] This could be because we failed to generate a TAT-HQA dataset that was comparable to the one used in the original paper.

Model	**FinQA**	**TAT-QA**	**HiTab**	**MultiHiertt**
CounterComp (operators)	**74.49**	**70.01**	**32.61**	**40.85**
CounterComp (operands)	68.98	70.80	28.88	37.67

Table A.6: Program accuracy of CounterComp versus a sampling strategy focused on operands. FinQANet was used for all experiments.

Algorithm 2 Training algorithm

```
Training data: $\{([Q||F]^{(i)}, S^{(i)})\}_{i=1}^{I}$
Parameters: $\lambda$
Model: *model*
// Create the indices for pos and neg samples
pos_index $\leftarrow \{\}$
neg_index $\leftarrow \{\}$
for $i \in \{1, \ldots, I\}$ do
    $O^{(i)} \leftarrow s_1^{(i)}, s_4^{(i)}, \cdots, s_{L-3}^{(i)}$
    add_to_index(pos_index, $O^{(i)}, i$)
// $p$ is the perturbed output and $l$ is the location of the perturbation
    for $p, l \in$ possible_perturbations($O^{(i)}$) do
        $j \leftarrow$ find_matching_sample($p$)
        add_to_index(neg_index, $O^{(i)}, (j, l)$)
    end for
end for
// Train (single epoch, non-batch version)
for $i \in \{1, \ldots, I\}$ do
    for $j \in \{1, 2, \cdots, 5\}$ do
// Basic model loss
        $\mathcal{L}^{(i)} \leftarrow$ loss(*model.forward*($[Q||F]^{(i)}$), $S^{(i)}$)
// Pos/neg sampling
        $O^{(i)} \leftarrow s_1^{(i)}, s_4^{(i)}, \cdots, s_{L-3}^{(i)}$
        pos_sample $\leftarrow$ sample(pos_index[$O^{(i)}$] $\setminus i$)
        neg_sample, $l \leftarrow$ sample(neg_index[$O^{(i)}$])
// Find candidate intervention spans
        $\mathcal{Q}^{(i)} \leftarrow$ find_intrvntn_span($i$)
        $\mathcal{Q}_{\text{pos}}^{(i)} \leftarrow$ find_intrvntn_span(pos_sample)
        $\mathcal{Q}_{\text{neg}}^{(i)} \leftarrow$ find_intrvntn_span(neg_sample)
// Calculate edit distances and loss
        $\text{NLD}_{\text{pos}}^{(i)} \leftarrow$ norm_edit_dist($\mathcal{Q}^{(i)}, \mathcal{Q}_{\text{pos}}^{(i)}$)
        $\text{NLD}_{\text{neg}}^{(i)} \leftarrow$ norm_edit_dist($\mathcal{Q}^{(i)}, \mathcal{Q}_{\text{neg}}^{(i)}$)
        $\alpha^{(i)} = 1 - |\text{NLD}_{\text{neg}}^{(i)} - \text{NLD}_{\text{pos}}^{(i)}|$
        $\text{pos\_dist}_j^{(i)} = ||\mathbf{h}_l^{(i)} - \mathbf{h}_{l,\text{pos}}^{(i)}||_2^2$
        $\text{neg\_dist}_j^{(i)} = ||\mathbf{h}_l^{(i)} - \mathbf{h}_{l,\text{neg}}^{(i)}||_2^2$
        $\mathcal{L}_{\text{triplet}_j}^{(i)} = \max\{\text{pos\_dist}_j^{(i)} - \text{neg\_dist}_j^{(i)} + \alpha^{(i)}, 0\}$
    end for
    $\mathcal{L}^{(i)} = (1 - \lambda)\mathcal{L}^{(i)} + \frac{\lambda}{5}\sum_{j=1}^{5} \mathcal{L}_{\text{triplet}_j}^{(i)}$
end for
$\mathcal{L} = \frac{1}{I}\sum_{i=1}^{I} \mathcal{L}^{(i)}$
*model.backward*($\mathcal{L}$)
```

Appendix B

Chapter 4 Appendices

B.1 AliGATr's experimental settings

For AligNet, we set the alignment parameter $\mathcal{D} = 0.01$. This means that if the horizontal or vertical distance between a pair of nodes is smaller than 1% of the width or height of the page, the two nodes are considered aligned (and thus adjacent).

Our GCN backbone is a 2-layer RGAT, implemented by the `Pytorch Geometric` library.[1] We use a 2-layer unidirectional LSTM [49] as the RNN module.

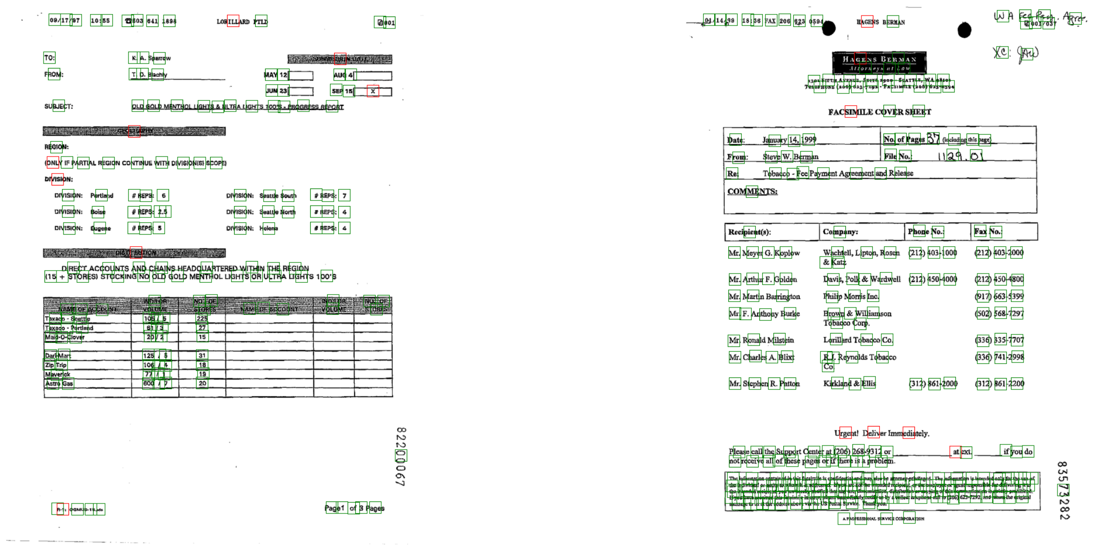
We use a sample of 1 million documents from the OCR-IDL dataset [13] to pre-train the model. During pre-training, we initialize the token embeddings using RoBERTaBASE[107]. We use a batch size of 1, a learning rate of $5\text{e}{-6}$, the AdamW optimizer [109] with $(\beta_1, \beta_2) = (0.9, 0.999)$, and train the model for 1 epoch. During fine-tuning for the KIE and RE tasks, we use a batch size of 16, learning rate of $1\text{e}{-5}$, the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, and train the model for 1000 epochs. We set the negative sampling rate for the co-distillation loss as 5.

B.2 Qualitative examples of AliGATr's performance

Figure B.1 shows the performance of AliGATr on the KIE task on two samples from the FUNSD dataset. As the figure shows, AliGATr struggles with tokens that do not have a clear alignment with other elements of a similar class. This can be attributed to AliGATr's weaker text backbone compared to other SotA models.

Figure B.2 shows the performance of AliGATr on the RE task on two samples from the FUNSD dataset. AliGATr recovers all edges that correspond to aligned elements. The performance is lower for elements that are not horizontally or vertically aligned. Notably, the rate of false negatives is higher than false positives.

[1] `https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.RGATConv.html`



(a) KIE results on form with tabular segments

(b) KIE results on sparse form

Figure B.1: KIE results on two samples from the FUNSD dataset. Green boxes show correct predictions and red boxes show incorrect predictions.

B.3 Constructing β-skeleton graphs

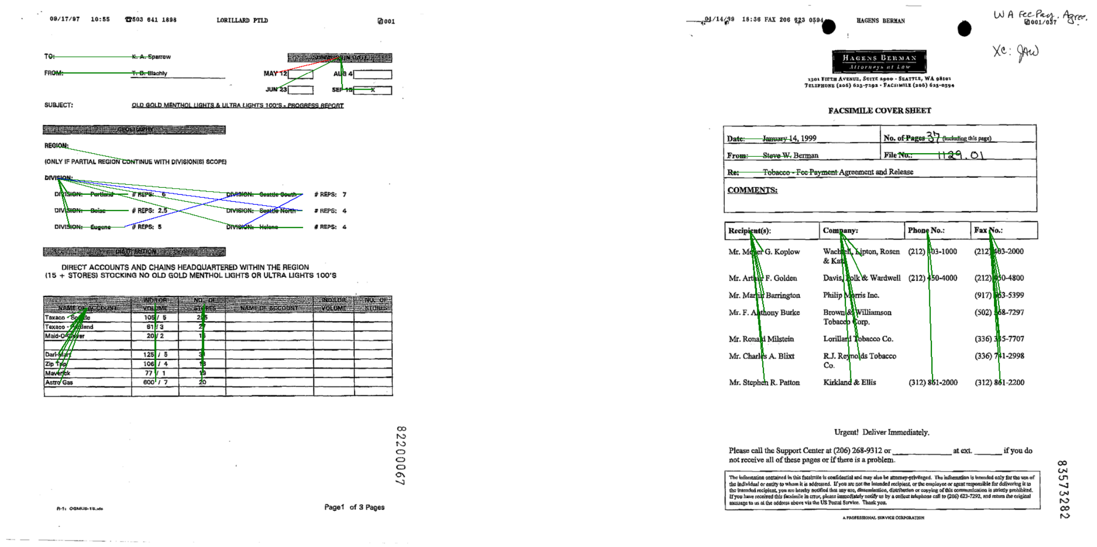
As mentioned in Section 4.2.1, the β-skeleton graph is favored in many graph-based form understanding models. Consistent with Lee et al. [89], we set $\beta = 1$, making our graph a Gabriel graph-—a subset of Delaunay triangulation [78]. Unlike typical point-based β-skeleton graphs, our approach involves bounding box β-skeleton graphs. We use each token's four coordinates (top-left, top-right, bottom-left, bottom-right) as vertices and employ Delaunay triangulation from `scipy.spatial`[2] to construct the graph, as shown in figure B.3a. We then remove all internal connections within a bounding box. While a strict β-skeleton graph would exclude any edges with vertices inside the circle formed by those edges, this results in excessive sparsity due to token proximity. To address this, we maintain all edges but simplify by collapsing the four corners to the center of each bounding box, as demonstrated in figure B.3b.

B.4 Community sensitive self-supervision

B.4.1 Community detection

The AligNet representation can be used to segment the page based on alignments, using a graph segmentation algorithm. These algorithms are designed to find cliques, partitions, or communities (i.e. locally dense segments) within the graph. Among such algorithms, the Leiden community detection method [178] is a particularly useful approach, because: 1) It focuses on maximiz-

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Delaunay.html`



(a) RE results on form with tabular segments

(b) RE results on sparse form

Figure B.2: RE results on two samples from the FUNSD dataset. Green links show correct predictions. Red links show false negatives. Blue links show false positives.

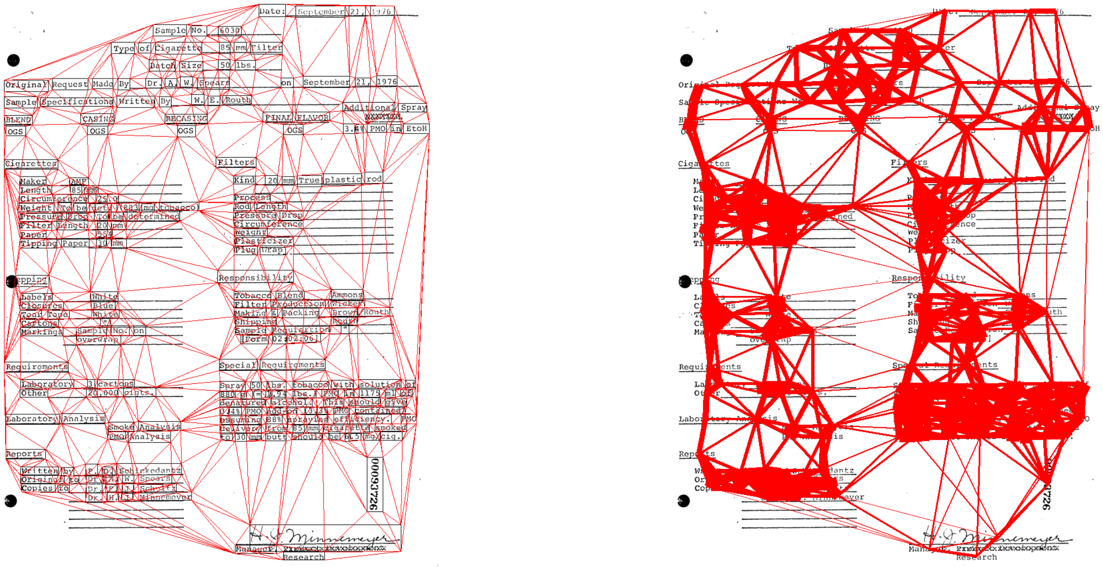
ing the modularity of a network, which is defined as the density of intra-community edges compared to inter-community edges. This is congruent with the segmentation objective in AligNet, since high-density areas of a page can indicate a segment (see Figure B.4c). 2) The greedy implementation of the Leiden method leads to log-linear complexity in most experimental settings [85], which offers a runtime advantage. 3) The recursive nature of the Leiden method exempts it from requiring a pre-determined number of communities. The algorithm stops when the overall modularity of the network can no longer be improved beyond a minimum threshold.

We use the implementation of the algorithm offered by the `NetworkX` python library[3]. Edges are weighted according to the following distance calculation:

$$w_{e_{ij}} = \frac{\mathcal{W}(e_{ij})}{\sqrt{(b_i^{\text{center}} - b_j^{\text{center}})^2 + (b_i^{\text{middle}} - b_j^{\text{middle}})^2}} \tag{B.1}$$

where $\mathcal{W}(e_{ij})$ is a weighing hyperparameter. This weighing scheme allows the Leiden algorithm to consider distance and proximity when identifying the segments. Once the algorithm converges, each node in the AligNet graph v_i is assigned a community label c_i.

[3] https://networkx.org/documentation/networkx-3.1/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html



(a) Point-based β-skeleton graph on bbox coordinates

(b) β-skeleton graph merging internal bbox connections

Figure B.3: Construction of a β-skeleton graph on a sample form. First, create a point-based β-skeleton graph with the 4 corners of each bounding box as vertices (a). Next, remove internal connections within each bounding box and merge the 4 vertices into the centroid (b). The width of the edges in (b) indicates the edge weight: shorter edges have higher weights.

B.4.2 Community-aware GAT

The classic GAT model [185] learns the representation of each node by convolving its original representation with those of its neighbors. In the GATv2 convolution [16], this is designed as:

$$\mathbf{h}_i' = \alpha_{i,i}\mathbf{\Theta}_s\mathbf{h}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}_t\mathbf{h}_j \tag{B.2}$$

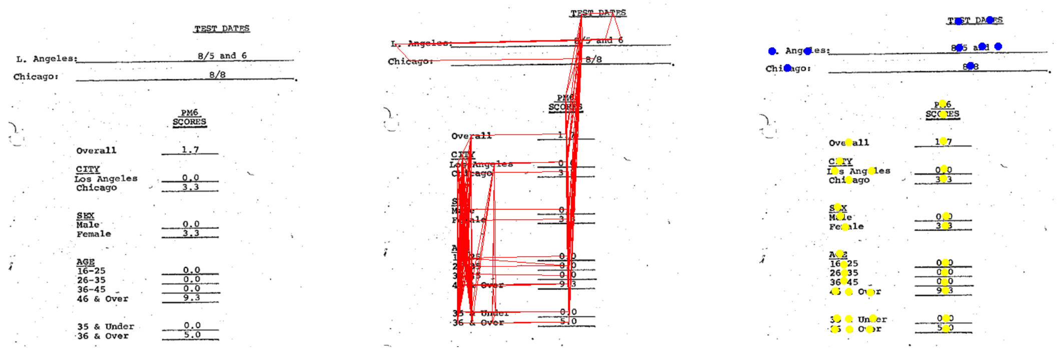
The attention score $\alpha_{i,j}$ is calculated as:

$$\alpha_{i,j} = \frac{\exp(\mathbf{f}^\top \mathrm{LeakyReLU}(\mathbf{\Theta}_s\mathbf{h}_i + \mathbf{\Theta}_t\mathbf{h}_j + \mathbf{\Theta}_e\mathbf{e}_{i,j}))}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp(\mathbf{f}^\top \mathrm{LeakyReLU}(\mathbf{\Theta}_s\mathbf{h}_i + \mathbf{\Theta}_t\mathbf{h}_k + \mathbf{\Theta}_e\mathbf{e}_{i,k})} \tag{B.3}$$

where $\mathbf{f}$ is an affine parameter, $\mathcal{N}(i)$ represents the set of nodes adjacent to x_i and $\mathbf{\Theta}_s$, $\mathbf{\Theta}_t$, and $\mathbf{\Theta}_e$ are weight parameters corresponding to source, target, and edge representations, respectively.

For Leiden community detection, we set $\mathcal{W}(e_{ij})$ to 1 for horizontal edges and to $\frac{1}{16}$ for vertical edges. This encourages the algorithm to prioritize the merging of nodes along horizontal edges, which leads to the creation of horizontally-aligned segments. This is consistent with the general reading order of English-language documents (left-to-right, then top-to-bottom), but can be adjusted for other languages.

In a similar fashion, a community detection algorithm can be used to segment the β-skeleton graph (see Figures B.7 and B.8).



(a) Raw form (b) AligNet graph (c) Leiden communities

Figure B.4: Visual illustration of how the AligNet representation can enable page segmentation. The example document is excerpted from FUNSD [63].

B.4.3 Graph Representations and Number of Communities

	Graph Structure	
# of communities	β-skeleton	AligNet
baseline	16.56	**19.53**
1	0.0	11.10
2	**20.50**	13.45
4	16.68	17.03
8	15.48	-
16	16.24	-

Table B.1: Ablation results on graph representations and community numbers. For the β-skeleton graph, 2 communities per document yield the best results. For the AligNet graph, the baseline with the Louvian algorithm's optimal community number performs best. All numbers reflect F1 performance on the FUNSD dataset.

In this section, we investigate the effect of splitting each page into a predetermined set of communities. Our ablation experiments aim to compare different graph representations, focusing on β-skeleton graphs and AligNet graphs. All experiments use $\beta = 1$ (Gabriel graph) and a very small pre-training dataset of 149 documents from FUNSD [63]. We also explored the effects of different community detection configurations using the Leiden method, focusing on variations in the resolution parameter γ and explicitly setting community sizes.

Adjusting the resolution parameter γ allowed us to control community detection granularity, impacting both community count and size. Higher γ values led to more but smaller communities. However, γ adjustments did not yield consistent results across different graph structures. Therefore, we predetermined the number of communities by initially assigning nodes to a set number of groups, allowing the algorithm to refine these into fixed community counts without exceeding the predefined limits.

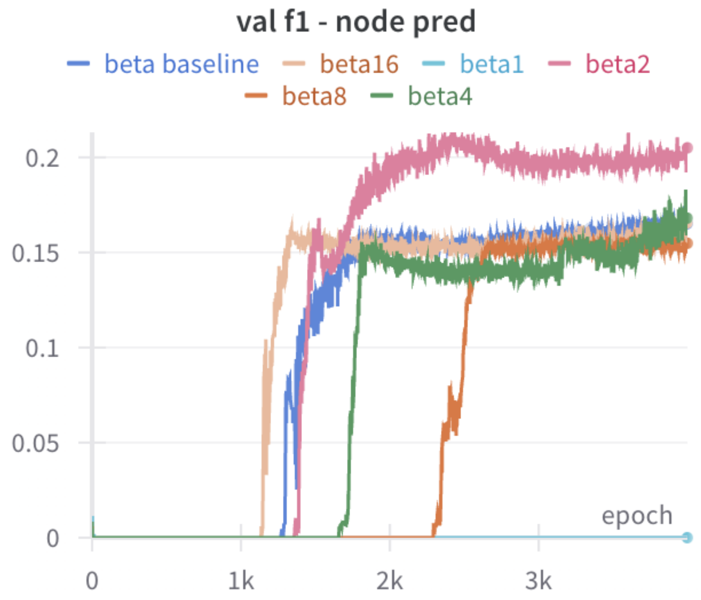
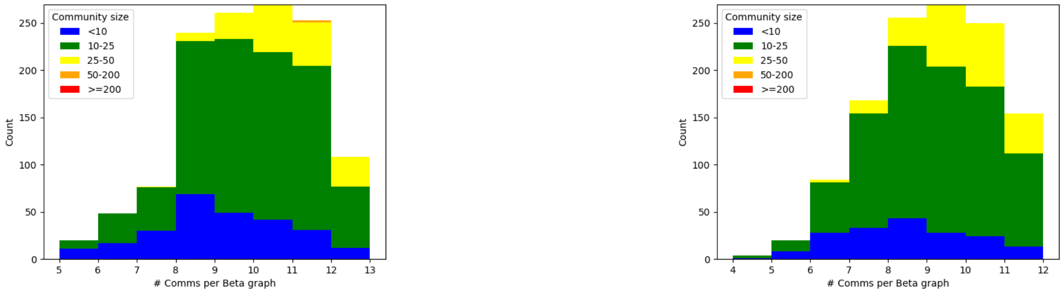


Figure B.5: Training curves for β-skeleton with different community sizes. The number after ”beta” in the legend indicates the number of communities per document. β-skeleton with 2 communities yields the best results. Graphs with 4 and 8 communities have lower convergence rates compared to the baseline and the graph with 16 communities, despite similar F1.

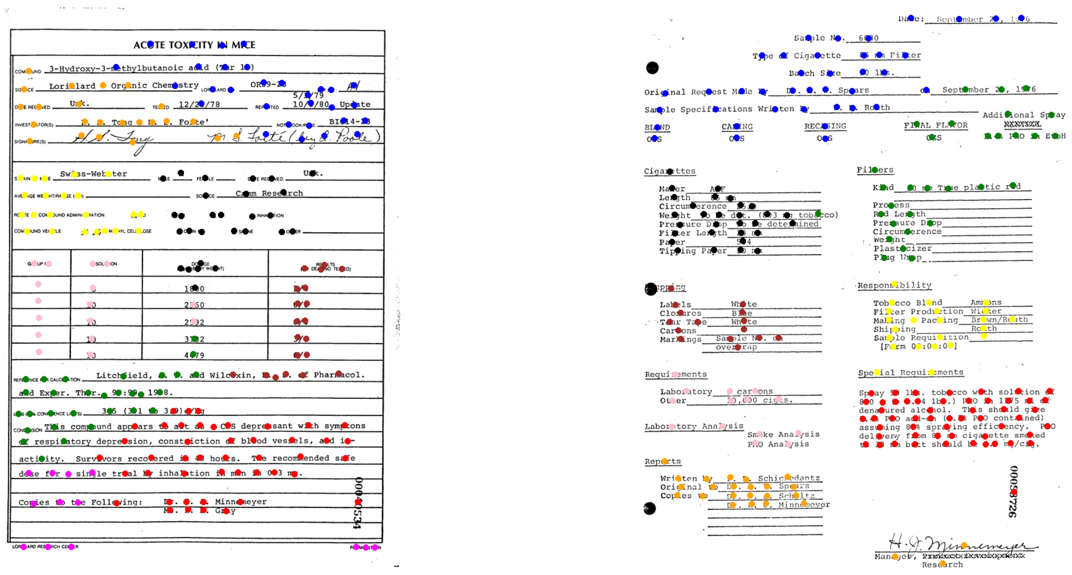


(a) β-skeleton community distribution, baseline configuration.

(b) β-skeleton community distribution, $\#$ of comm = 16

Figure B.6: Cumulative counts for community size for β-skeleton graphs. In (a), for graphs with 5 communities (left-most column), approximately half of them have less than 10 nodes (blue segment at the bottom), while the other half have 10-25 nodes (green segment). The typical community size for β-skeleton graphs is 10-25 nodes. In (b), when explicitly setting the maximum community size to 16, the distribution trend is similar to (a).

For the β-skeleton graph, setting all nodes into a single community prevented the model from converging. As shown in Figure B.5, models with two communities achieved higher and more stable F1 scores throughout the epochs. Conversely, increasing the number of communities to



(a) β-skeleton communities for a form with tabular segments

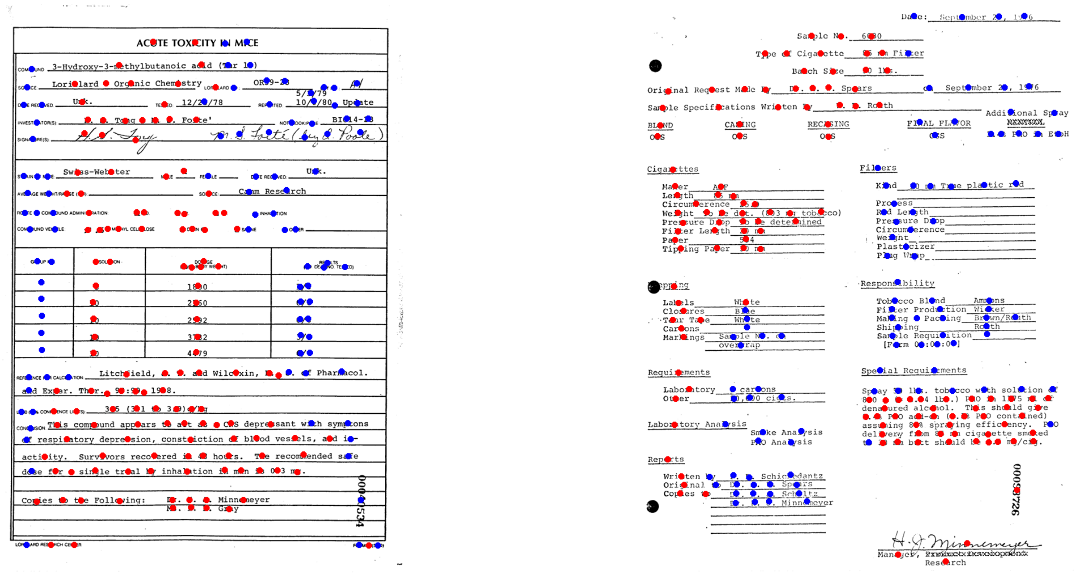
(b) β-skeleton communities for form with nested segments

Figure B.7: β-skeleton communities baseline. In uniformly dense documents, communities are not properly separated (a), whereas distinct communities are formed in less dense and more structured documents (b).

4, 8, or 16 generally decreased performance. Notably, models with 4 or 8 communities showed slower convergence rates, whereas configurations with 16 communities unexpectedly improved convergence compared to the previous two. Further analysis of community size distributions, shown in Figure B.6, reveals that setting the community number to 16 aligns the distribution closely with the baseline model, resulting in similar performance trends. Moreover, Figure B.6 also indicates that the maximum viable number of communities for the β-skeleton graph is 12, highlighting the graph's limitations due to sparsity.

For AligNet, the results in Table B.1 show a different pattern compared to the β-skeleton graph. The baseline model, which uses the Leiden algorithm to determine the optimal community numbers, achieves the best F1 scores. However, explicitly setting a lower number of communities results in lower F1 scores. The lowest F1 score is observed when all nodes are grouped into a single community. These findings suggest that AligNet performs optimally with multiple, smaller-sized communities.

Our findings indicate that community information enhances modeling for both AligNet and β-skeleton graphs, each benefiting from different community configurations. The β-skeleton graph performs optimally with larger communities, effectively utilizing extensive neighboring information, while the AligNet graph is more effective with finer community granularity. For the β-skeleton graph, smaller communities do not ensure accurate separation into distinct blocks in uniformly dense documents, as shown in Figure B.7a. Conversely, utilizing larger communities reduces the focus on smaller clusters and enhances the separation of specific tokens like "questions" and "answers", thereby improving the task performance as shown in Figure B.8.



(a) β-skeleton communities with #comms = 2 for a form with tabular segments

(b) β-skeleton communities with #comms = 2 for a form with nested segments

Figure B.8: β-skeleton communities with number of communities = 2. The visualizations show the communities generally separating out the ”question” type token on the left and the ”answer” type tokens in the middle.

B.5 Order sensitive edge representations

Building on the findings of Lee et al. [87], we explored the impact of reading order on graph structures in our ablation experiments. All experiments in this section are base on a small pre-training dataset of 149 documents from FUNSD [63].

Raw Distance: We modified our approach by using raw distances instead of the original edge definition shown in the edge representation described in Section 4.2.2. The distance between nodes x_i and x_j is represented as:

$$\mathbf{e}_{i,j} = [b_i^{\text{left}} - b_j^{\text{left}}, b_i^{\text{right}} - b_j^{\text{right}}, b_i^{\text{top}} - b_j^{\text{top}}, b_i^{\text{bottom}} - b_j^{\text{bottom}}] \tag{B.4}$$

We hypothesize that this raw distance can implicitly convey reading order, with negative values suggesting x_i precedes x_j, and positive values indicating the reverse.

Order-Sensitive Edge Label: We initially defined alignment edge labels as

$$\exists c \in \{\text{left}, \text{center}, \text{right}, \text{top}, \text{middle}, \text{bottom}\}$$

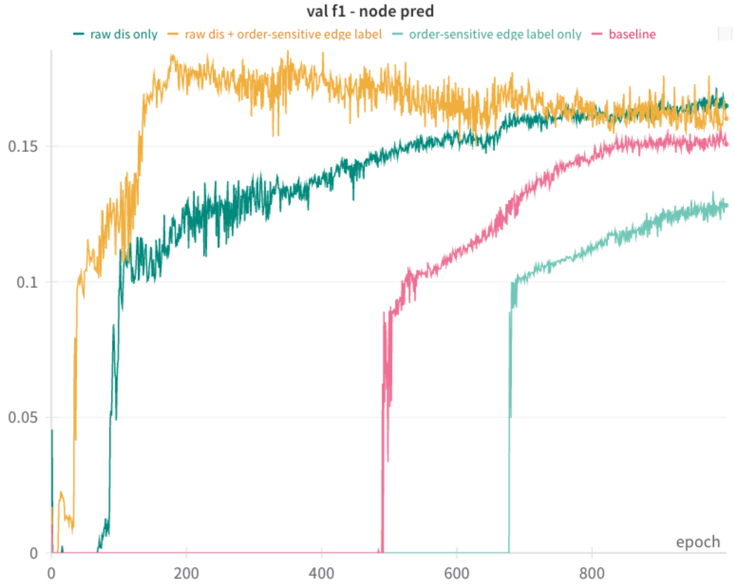


Figure B.9: Training curves comparing different edge representations show that incorporating raw distance improves model performance and accelerates convergence compared to the baseline. Additionally, utilizing order-sensitive labels along with raw distance leads to even faster convergence. However, using order-sensitive labels alone results in slower convergence.

as described in Section 4.2.2. For our ablation experiments, we expanded these into twelve labels:

$$
\begin{aligned}
&\exists c \in \\
&\{\text{left}_{pre}, \text{center}_{pre}, \text{right}_{pre}, \\
&\text{top}_{pre}, \text{middle}_{pre}, \text{bottom}_{pre}, \\
&\text{left}_{post}, \text{center}_{post}, \text{right}_{post}, \\
&\text{top}_{post}, \text{middle}_{post}, \text{bottom}_{post}\}
\end{aligned}
$$

The label is determined by the summation of vectors in B.4: negative sums result in one of the first six labels (which x_i precedes x_j), while positive sums assign one of the latter six, indicating x_i follows x_j. This adjustment aims to further encode the reading order into the graph.

	w/ Raw Distance	w/o Raw Distance
Order-invariant labels	16.49	15.08
Order-sensitive labels	**18.29**	12.82

Table B.2: Ablation study results for edge representations in AligNet, showing F1 performance on the FUNSD dataset. Utilizing raw distance and explicit order-sensitive labels improves model performance.

Incorporating raw distance as implicit order-sensitive edge representations improves model performance, as shown in Table B.2 and Figure B.9. However, adding explicit order-sensitive edge labels did not consistently enhance performance. The edge labels did improve convergence when combined with raw distance, but using them alone resulted in slower convergence.

B.6 Representing edge types

	Edge Types							
#	**horizontal short**	**horizontal long**	**vertical short**	**vertical long**	**beta horizontal**	**beta vertical**	**beta other**	**F1**
1	✓							**42.99**
2		✓						38.21
3			✓					30.05
4				✓				36.82
5					✓			39.17
6						✓		37.74
7							✓	35.99
8	✓	✓						41.24
9	✓		✓					34.53
10		✓	✓					29.38
11	✓			✓				35.46
12	✓				✓			36.92
13	✓				✓	✓	✓	37.92
14					✓	✓	✓	35.51

Table B.3: Ablation results on edge types. The baseline F1 score is 35.81, using all edges in AligNet. Experiments 1-7 use single edge types, with horizontal short edges performing best. Experiments 8-11 test combinations of edge types for AligNet. Experiments 12-14 evaluate the effect of adding horizontal short edges to the β-skeleton graph.

Of the two graph structures, we also perform ablation studies to determine which edge types are useful for the task. In this section, all experiments use segment loss instead of community loss, and are based on a small pre-training dataset consisting of 149 documents from FUNSD [63]. In AligNet, we classified edges into four categories: horizontal-long, horizontal-short, vertical-long, and vertical-short. We set a threshold $\lambda = 0.3$ for short edges, including those shorter than 30% of the page width or height, and $\lambda = 0.5$ for long edges, which are longer than 50% of the page dimensions. Edges not meeting these criteria were excluded. For the β-skeleton graph, which primarily comprises short edges, we categorized edges into three groups: horizontal, vertical, and others, using the same threshold criteria as AligNet for orientation determination.

In our analysis, we used the AligNet graph as the baseline for comparison. The results, depicted in Table B.3 (Experiment 1-7), indicate that horizontal short edges significantly outperform all other types. Vertical short edges from AligNet notably reduced performance relative to the baseline, while vertical long edges and horizontal long edges showed performance similar to the baseline. For the β-skeleton graph, horizontal edges provided a slight improvement over the baseline.

We further tested combinations of edge types, maintaining the same experimental settings and considering the union of overlapping edge types. As also shown in Table B.3 (Experiment

8-11), the best results within the AligNet graph were achieved by combining horizontal short and horizontal long edges, with performance trends similar to those of horizontal short edges alone. Conversely, combinations of horizontal long and vertical short edges yielded the poorest results, indicating their limited utility. In the β-skeleton graph (Experiment 12-14), adding horizontal short edges enhanced performance. Those results show the importance of horizontal short edges in effectively connecting segment information in this task.

Appendix C

Chapter 5 Appendices

C.1 A $\beta-$skeleton based grounding algorithm

Algorithm 3 describes a possible way to ground a model output O within a page, without apriori access to the reading order. First, a page is represented by a β-skeleton graph, similar to Lee et al. [87]. Next, the first and last tokens of O are matched to the page by finding all nodes (i.e. tokens) on the graph that have a Levenshtein similarity to the first and last token, beyond a threshold T. Lastly, all possible paths between such nodes are found, and the path with the highest NLS to O is selected as the matching path.

A threshold can be set on the score of the best matching path S, below which the path is considered a mismatch and therefore no effective matches are found on the page, e.g. in cases when the model has hallucinated the output.

This algorithm ensures that any path that is matched to O is within a contiguous 2-D walk on the page, without the need for information related to reading order. A major downside of this algorithm is its quartic time complexity, which can be improved by caching partial paths. Nevertheless, we decided to use a simpler algorithm that relies on the reading order provided by OCR tools.

C.2 Additional experimental details for SMuDGE

C.2.1 Merging bounding boxes

A sequence of bounding boxes can be merged by finding the left-most, top-most, right-most, and bottom-most corners in the sequence in order to create a new bounding box. If all bounding boxes in the sequence form a contiguous segment, merging them would yield their union. However, if the bounding boxes are in disparate locations, this simple merging algorithm will not yield their union, and will cover additional areas. As an example, if a ground truth answer spans two lines, covering the second half of one line and the first half of the next, the merging algorithm will create a bounding box that covers both lines in full. Despite this limitation, we use this algorithm because we are only interested in measuring the distance between the resulting bounding boxes based on their centroids.

C.2.2 Determining the semantic type of the predicted answer

To classify a string of characters s as numeric, textual, or hybrid, we follow the below algorithm:

1. If every character in s is a digit, then s is numeric.
2. If every character in s is alphabetical, then s is textual.
3. Otherwise s is hybrid.

Note that this simple algorithm renders a large portion of strings such as “1,700” or “(8)” as hybrid. This is not detrimental to SMuDGE, as it still favors the accuracy of numbers against non-numeric characters by a factor of 10 to 1.

Note that hybrid strings are split into numeric sequences and non-numeric sequences, e.g. “1,700” is split into “1700” and “,” and each part is evaluated separately before being combined in the weighted harmonic mean.

C.2.3 Tuning the weights for the numeric score and the text score

We tuned the weight of num_score$_i$ against str_score$_i$ by testing $\{1, 10, 100, 1000\}$. The tuning was performed on a subsample of 100 hybrid answers from the DocVQA validation set.

C.2.4 Calculating volatility

We use the standard definition of volatility as scaled standard deviation:

$$\text{vol}([x_1, \cdots, x_T]) = \text{std}([x_1, \cdots, x_T])\sqrt{T} \tag{C.1}$$

C.3 Determining the types of questions in DocVQA

To determine the type of each question, we passed the following information to GPT-4o: 1) The document image. 2) The question. 3) The ground truth answer, as provided by the dataset. 4) A prompt, asking the model to determine the context from which the answer was extracted.

You can see an example prompt below:

Question: What is the extension number?
Answer: 5177
The above question was answered based on the document attached. What do you think best describes the context from which the answer was extracted? Select one of the below options. Simply return the correct option without any explanation.

1. **Figure/Diagram**
2. **Form**
3. **Table/List**
4. **Layout**
5. **Free_text**
6. **Image/Photo**
7. **Handwritten**
8. **Yes/No question**
9. **Other**

The experiment ran on September 7th, 2024. The agreement rate with the DocVQA validation set was 69.5%.

C.4 Extended leaderboard analysis

Figures C.1 to C.3 show the reranking analysis for MP-DocVQA, InfographicVQA, and DUDE benchmarks, respectively. As with Figure 5.2, our composite score has been calculated with $\alpha = 0.25$.

C.5 Extended question type analysis for DocVQA

Figure C.4 shows how the top 10 models on the DocVQA leaderboard would be reranked if our score was used to evaluate them, broken down by question types.

C.6 Answer type analysis for DocVQA

Figure C.5 shows how the top 10 models on the DocVQA leaderboard would be reranked if our score was used to evaluate them, broken down by answer types.

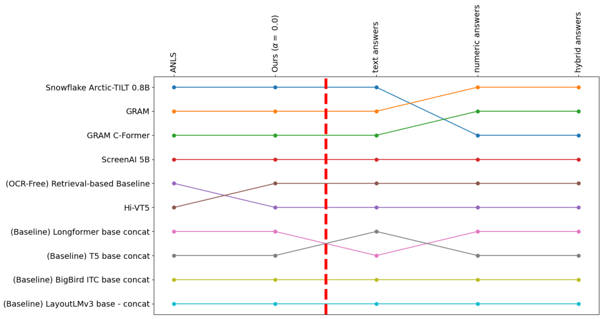


Figure C.1: MP-DocVQA leaderboard.

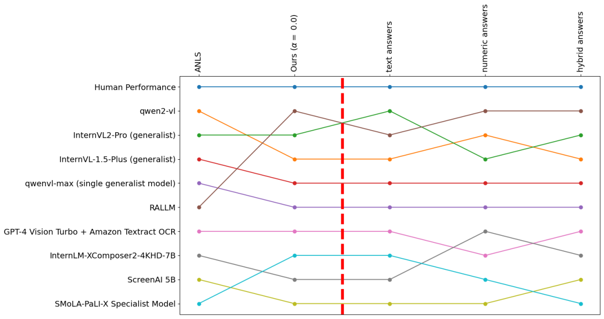


Figure C.2: InfographicVQA leaderboard.

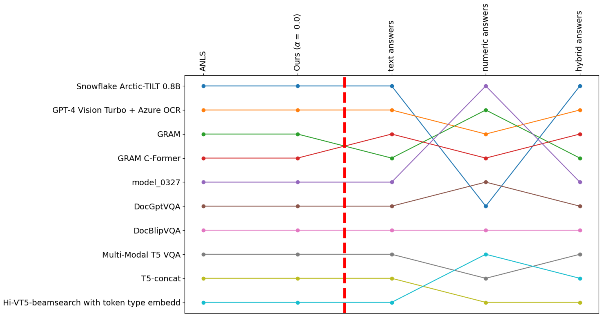
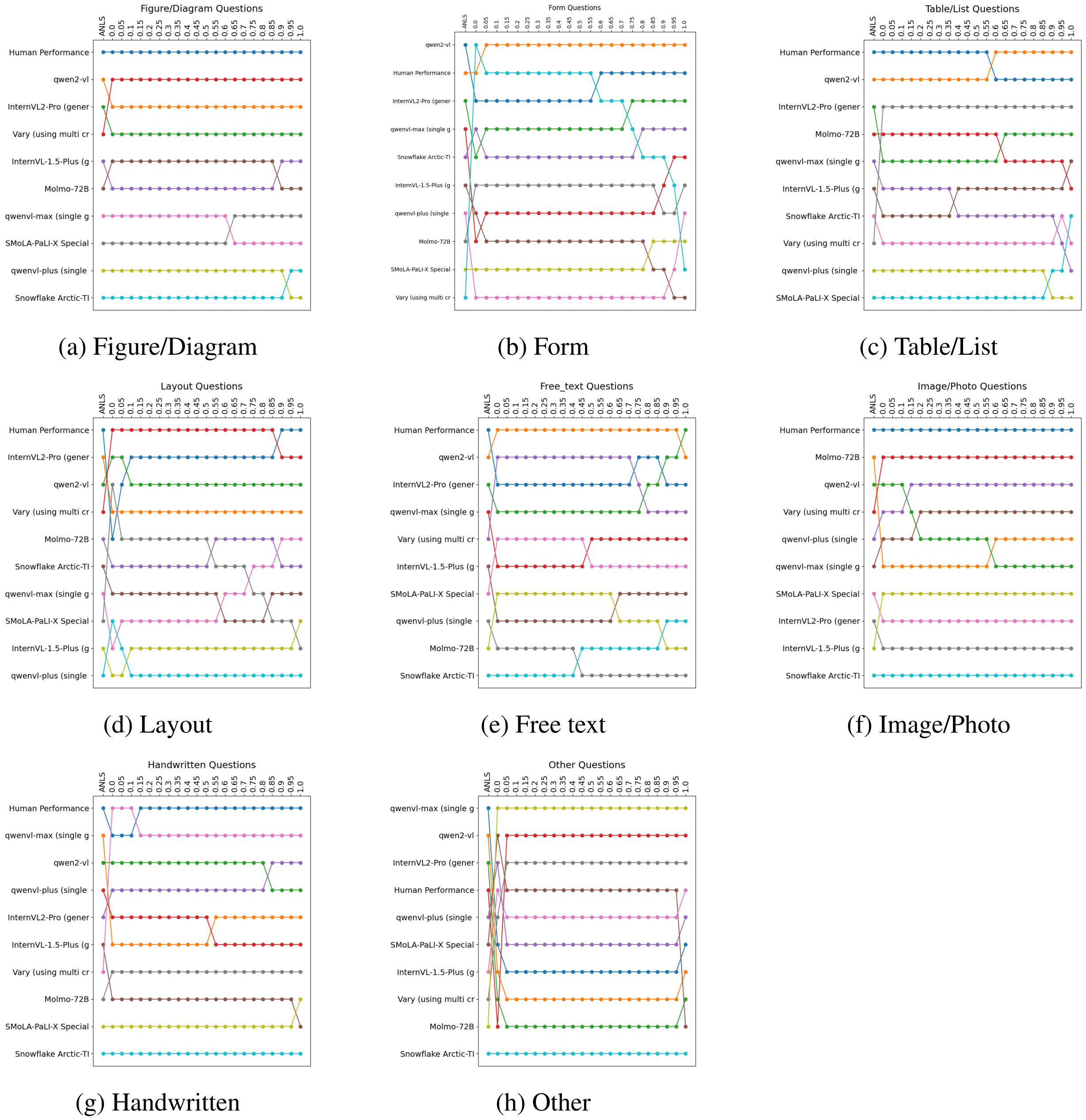


Figure C.3: DUDE leaderboard.



(a) Figure/Diagram

(b) Form

(c) Table/List

(d) Layout

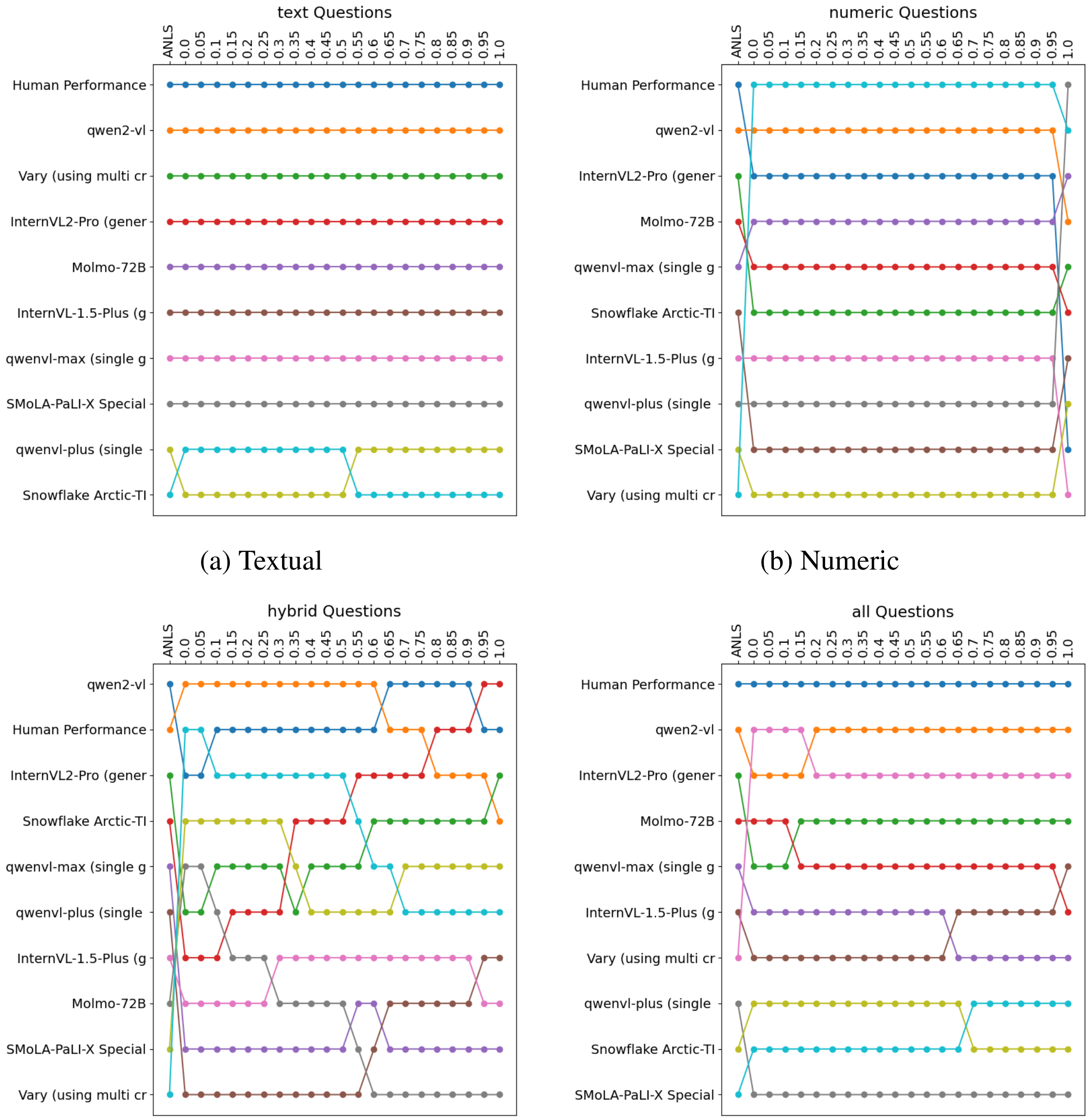
(e) Free text

(f) Image/Photo

(g) Handwritten

(h) Other

Figure C.4: The impact of our score on the ranking of the top 10 models on the DocVQA benchmark, broken down by question type.



(a) Textual

(b) Numeric

(c) Hybrid

(d) All

Figure C.5: The impact of our score on the ranking of the top 10 models on the DocVQA benchmark, broken down by answer type.

C.7 Answer type analysis for MP-DocVQA

Figure C.6 shows how the top 10 models on the MP-DocVQA leaderboard would be reranked if our score was used to evaluate them, broken down by answer types.

C.8 Answer type analysis for InfographicVQA

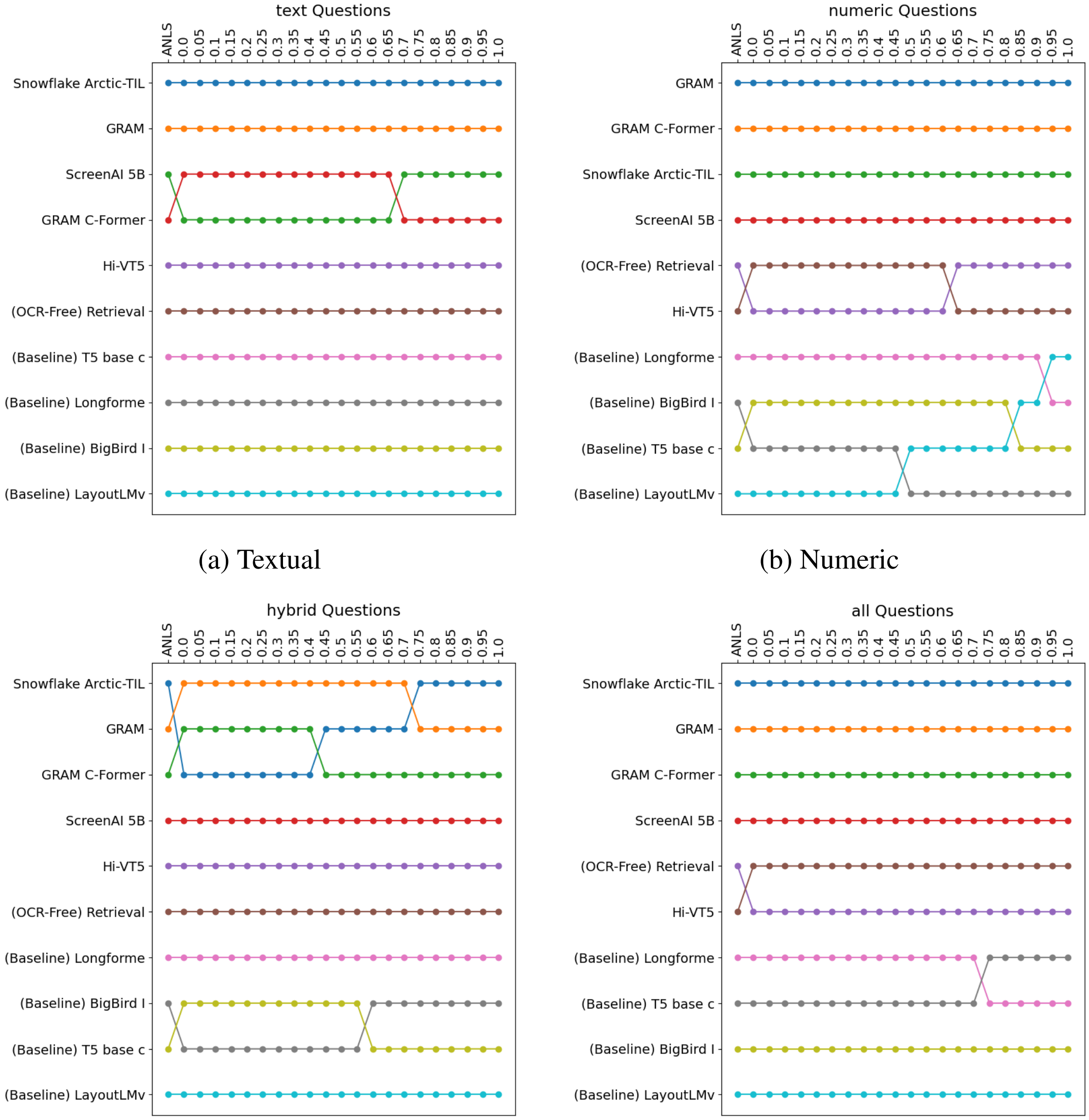
Figure C.7 shows how the top 10 models on the InfographicVQA leaderboard would be reranked if our score was used to evaluate them, broken down by answer types.

C.9 Answer type analysis for DUDE

Figure C.8 shows how the top 10 models on the DUDE leaderboard would be reranked if our score was used to evaluate them, broken down by answer types.

C.10 Correlation between answer types and original ranking

Figure C.9 shows the correlation between the ranking of each leaderboard and the ranking produced by SMuDGE at various various for α, broken down by the type of answer.



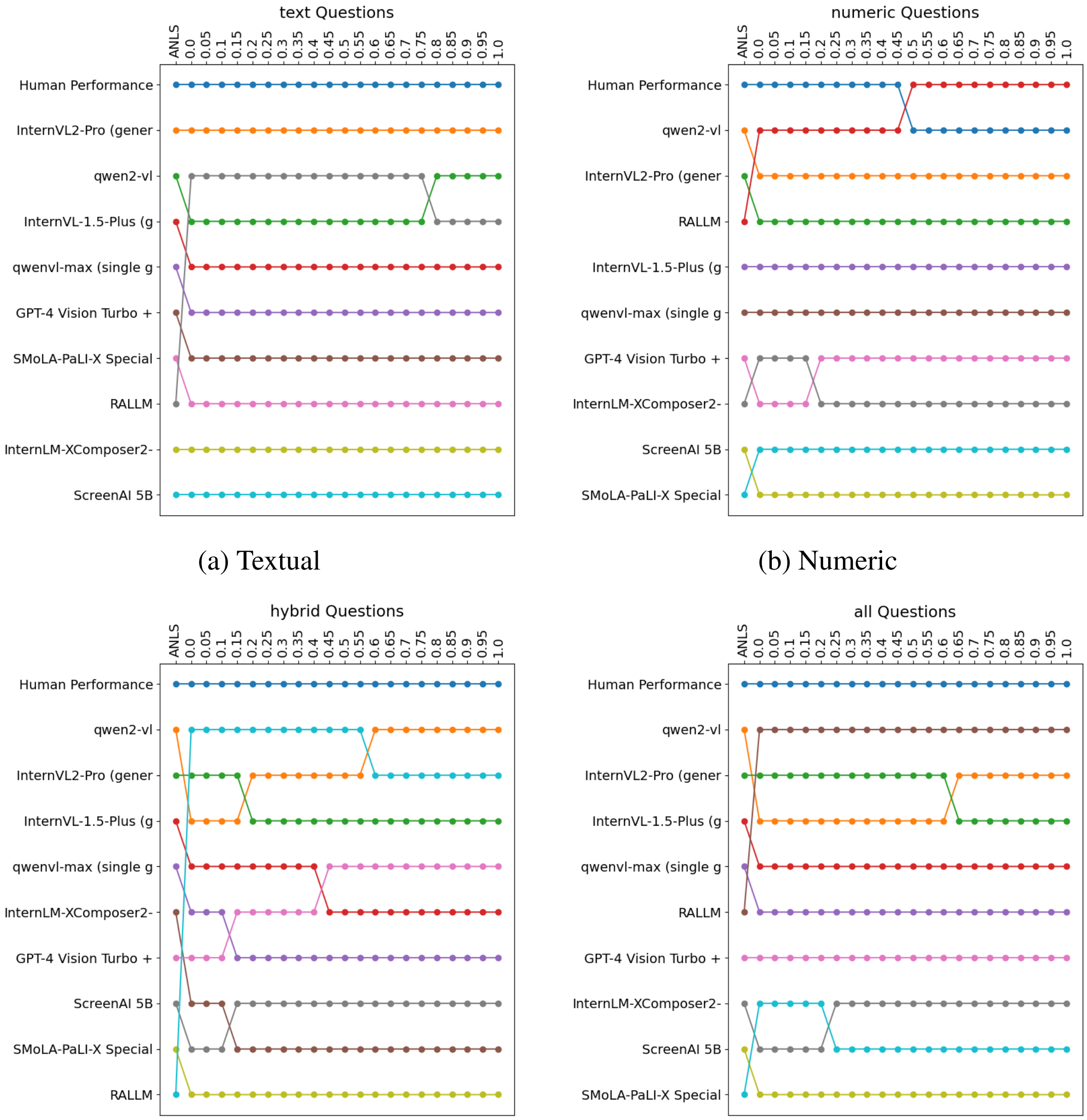
(a) Textual

(b) Numeric

(c) Hybrid

(d) All

Figure C.6: The impact of our score on the ranking of the top 10 models on the MP-DocVQA benchmark, broken down by answer type.



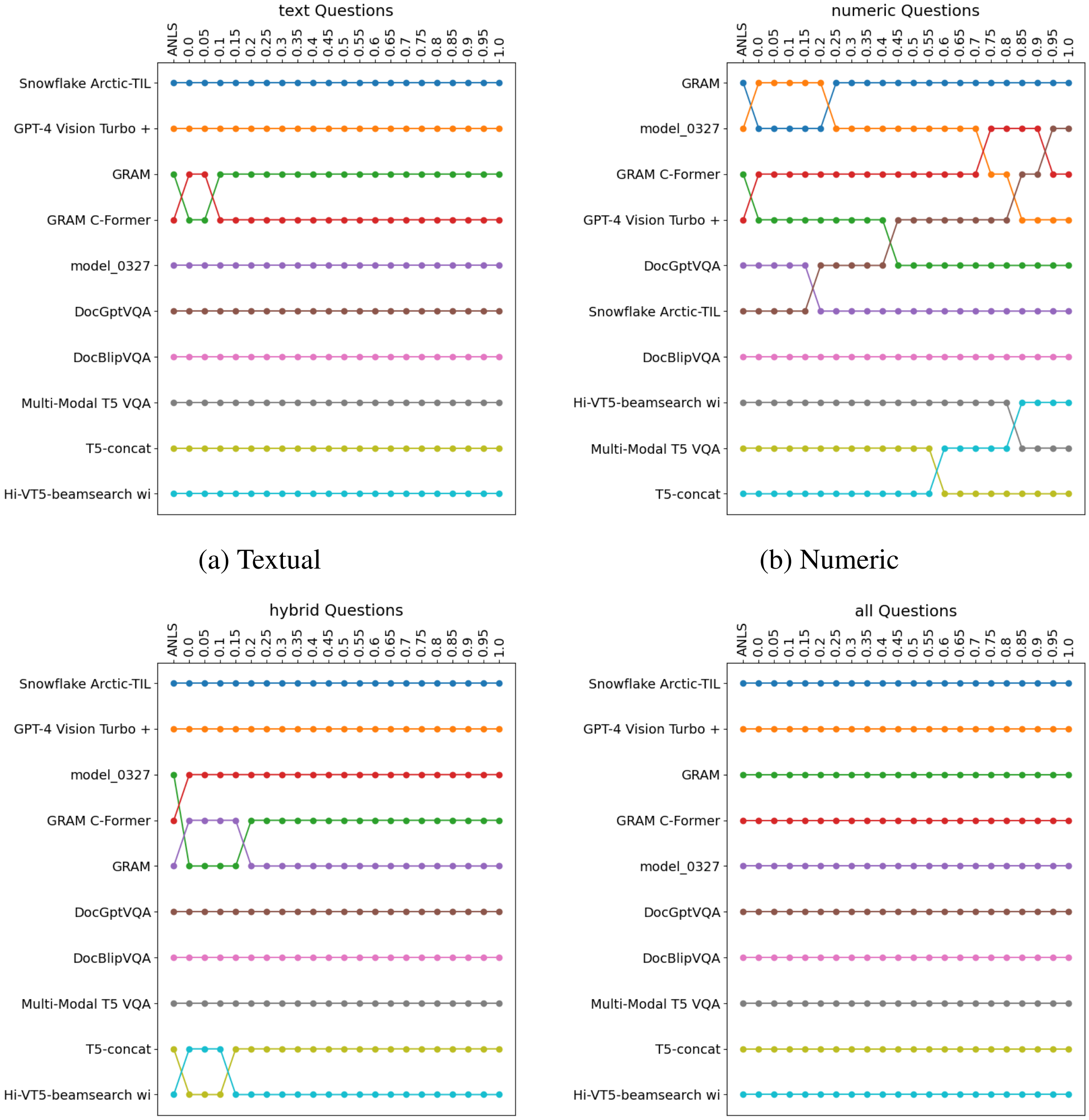
(a) Textual

(b) Numeric

(c) Hybrid

(d) All

Figure C.7: The impact of our score on the ranking of the top 10 models on the InfographicVQA benchmark, broken down by answer type.



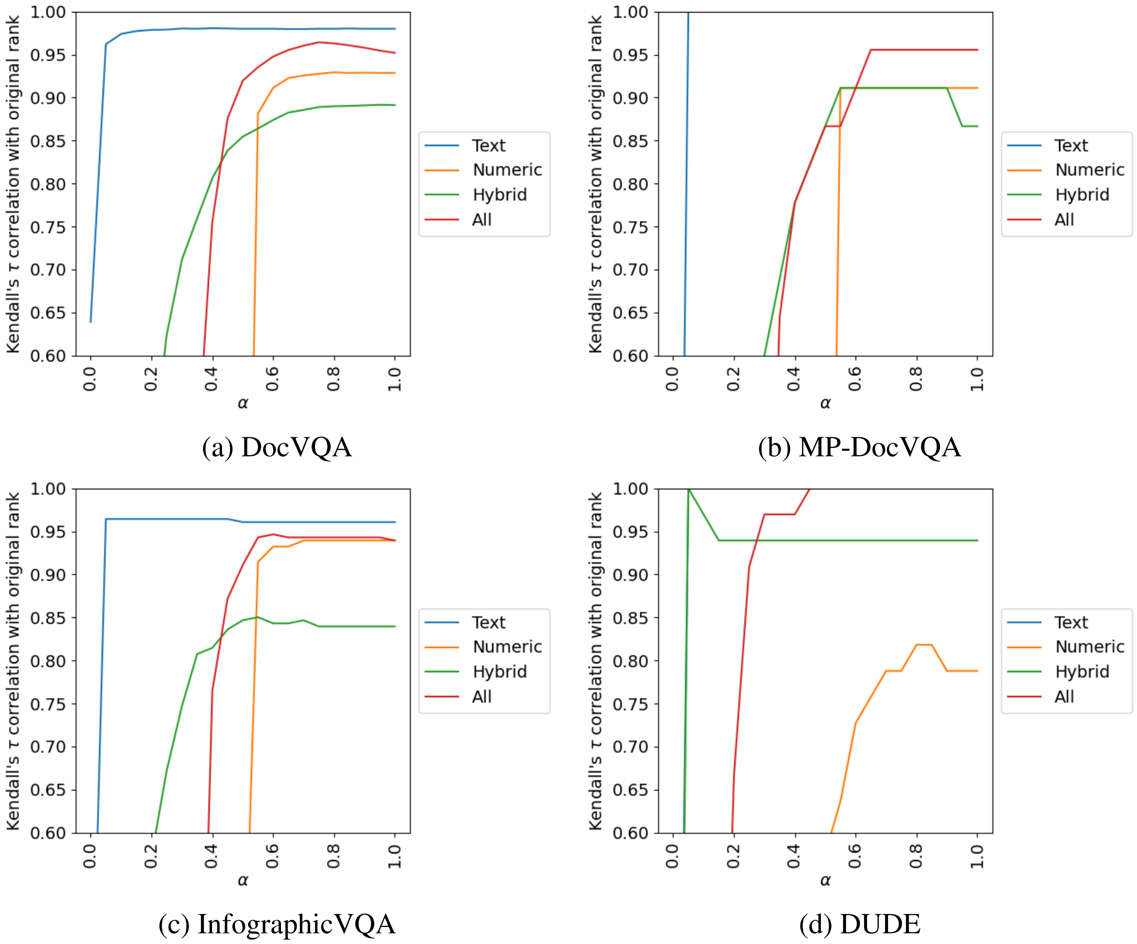
(a) Textual

(b) Numeric

(c) Hybrid

(d) All

Figure C.8: The impact of our score on the ranking of the top 10 models on the DUDE benchmark, broken down by answer type.



(a) DocVQA

(b) MP-DocVQA

(c) InfographicVQA

(d) DUDE

Figure C.9: Kendall's τ correlation between different α settings and the original ranking of each benchmark, broken down by the type of answers.

Algorithm 3 β-skeleton walk for placing a sequence of tokens within a page.

```
// $\beta$-skeleton representation of a page
Input: $G = (N, V)$
// Matching target: a sequence of tokens
Input: $O = o_1, o_2, \cdots, o_n$
// Threshold for token similarity
Input: $T$
// Best path on the graph that matches O
Output: $P$
// The similarity of the best path to O
Output: $S$
// Create empty indices of all possible paths over the graph, starting from $o_1$ ending in $o_n$.
$p_s \leftarrow \{\}$
$p_e \leftarrow \{\}$
for $i \in \{1, \ldots, |N|\}$ do
    $s_{i1} = \text{NLS}(N_i, o_1)$
    $s_{in} = \text{NLS}(N_i, o_n)$
    if $s_{i1} > T$ then
        append$(p_s, n_i)$
    end if
    if $s_{in} > T$ then
        append$(p_e, n_i)$
    end if
// Search all possible paths and select the one with the highest score
    for $p_j \in p_s$ do
        for $p_k \in p_e$ do
            for path $\in$ paths$(p_j \rightarrow p_k)$ do
                if NLS$(\text{path}, O) > S$ then
                    $S \leftarrow \text{NLS}(\text{path}, O)$
                    $P \leftarrow$ path
                end if
            end for
        end for
    end for
end for
```

Bibliography

[1] Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-09-15. 76, 92

[2] Tesseract version 5.0.0. `https://github.com/tesseract-ocr/tesseract`. Accessed: 2021-12-10. 53

[3] Apple intelligence foundation language models, 2024. URL `https://arxiv.org/abs/2407.21075`. 84

[4] Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. Numerical reasoning in machine reading comprehension tasks: are we there yet? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9643–9649, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.759. URL `https://aclanthology.org/2021.emnlp-main.759`. 13

[5] Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.744. URL `https://aclanthology.org/2024.acl-long.744`. 77

[6] 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL `https://aclanthology.org/N19-1245`. 4

[7] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 17

[8] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. Giving BERT a calculator:

Finding operations and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5947–5952, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1609. URL `https://aclanthology.org/D19-1609`. 24

[9] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 993–1003, October 2021. 6, 17, 18, 21, 24, 47, 59

[10] Petr Babkin, William Watson, Zhiqiang Ma, Lucas Cecchi, Natraj Raman, Armineh Nourbakhsh, and Sameena Shah. Bizgraphqa: A dataset for image-based inference over graph-structured diagrams from business domains. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2691–2700, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591875. URL `https://doi.org/10.1145/3539618.3591875`. 90

[11] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 6, 18, 83, 90

[12] Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/D12-1091`. xviii, 32, 44, 67

[13] Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset, 2022. 67, 109

[14] Łukasz Borchmann, Michał Pietruszka, Wojciech Jaśkowski, Dawid Jurkiewicz, Piotr Halama, Paweł Józiak, Łukasz Garncarek, Paweł Liskowski, Karolina Szyndler, Andrzej Gretkowski, et al. Arctic-tilt. business document understanding at sub-billion scale. *arXiv preprint arXiv:2408.04632*, 2024. 17, 18, 19

[15] Łukasz Borchmann, Michał Pietruszka, Wojciech Jaśkowski, Dawid Jurkiewicz, Piotr Halama, Paweł Józiak, Łukasz Garncarek, Paweł Liskowski, Karolina Szyndler, Andrzej Gretkowski, et al. Arctic-tilt. business document understanding at sub-billion scale. *arXiv preprint arXiv:2408.04632*, 2024. 80

[16] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=F72ximsx7C1`. 112

[17] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*, 2019. 62

[18] TrustBank CBS. Seamless solution for customer onboarding and kyc compliance. `https://www.trustbankcbs.com/`

solutions-Core-banking-software-customer-onboarding.html. Accessed: 2024-02-01. 1, 92

[19] Chase Media Center. Consumers are using banking apps for more than transactions, new chase study finds, 2024. URL https://media.chase.com/news/consumers-are-using-banking-apps-for-more-than-transactions-new-chase-study-finds. 1

[20] Hao Chen, Rui Xia, and Jianfei Yu. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.24. URL https://aclanthology.org/2021.emnlp-main.24. 38

[21] Justin Chih-Yao Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, and Mohit Bansal. Symbolic mixture-of-experts: Adaptive skill-based routing for heterogeneous reasoning. *arXiv preprint arXiv:2503.05641*, 2025. 92

[22] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.91. URL https://aclanthology.org/2020.findings-emnlp.91. 12, 24

[23] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 6

[24] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 18, 83

[25] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 80

[26] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300. xiii, xviii, 1, 4, 5, 8, 12, 13, 23, 26, 27, 29, 30, 32, 35, 36, 37, 39, 42, 46, 100

[27] Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.78. URL `https://aclanthology.org/2022.acl-long.78`. 1, 4, 5, 12, 29, 35, 42

[28] NRZ SERVICER ADVANCE RECEIVABLES TRUST CS. Amended and restated indenture, 2013. URL `https://www.sec.gov/Archives/edgar/data/1520566/000119312513481955/d648867dex43.htm`. xvii, 3, 7

[29] Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 18

[30] Brian L. Davis, B. Morse, Scott D. Cohen, Brian L. Price, and Chris Tensmeyer. Deep visual template-free form parsing. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 134–141, 2019. 49, 53

[31] Brian L. Davis, B. Morse, Brian L. Price, Chris Tensmeyer, and Curtis Wigington. Visual fudge: Form understanding via dynamic graph editing. In *IEEE International Conference on Document Analysis and Recognition*, 2021. URL `https://api.semanticscholar.org/CorpusID:234763397`. 16, 17, 18, 21, 24, 47, 53, 54, 55, 60

[32] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, James Park, Reza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. 2024. URL `https://molmo.allenai.org/paper.pdf`. 80, 84, 92

[33] Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 407–426, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.23. URL `https://aclanthology.org/2024.findings-acl.23/`. 92

[34] 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. 17

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-

training of deep bidirectional transformers for language understanding. 2018. URL https://arxiv.org/abs/1810.04805. 48, 50, 53

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 59

[37] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[38] Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. Assessing composition in sentence vector representations. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1152. 91

[39] Sorouralsadat Fatemi and Yuheng Hu. Enhancing financial question answering with a multi-agent reflection framework. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, page 530–537, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400710810. doi: 10.1145/3677052.3698686. URL https://doi.org/10.1145/3677052.3698686. 92

[40] Daniel Peter Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv e-prints*, page arXiv:2007.08970, 2020. 13

[41] Antoine Gorceix, Bastien Le Chenadec, Ahmad Rammal, Nelson Vadori, and Manuela Veloso. Learning mathematical rules with large language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=tIlDF5B6T4. 86

[42] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. Unidoc: Unified pretraining framework for document understanding. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=UMcd6l1msUK. 59

[43] Shijie Han, Changhai Zhou, Yiqing Shen, Tianning Sun, Yuhua Zhou, Xiaoxia Wang, Zhixiao Yang, Jingshu Zhang, and Hongguang Li. Finsphere: A conversational stock analysis agent equipped with quantitative tools based on real-time database, 2025. URL https://arxiv.org/abs/2501.12399. 92

[44] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*. 18, 48, 52

[45] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015. 65, 88

[46] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6,

17

[47] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.398. URL `https://aclanthology.org/2020.acl-main.398`. 13, 14, 21, 24, 37, 47

[48] Nicolas Hiebel, Olivier Ferret, Karen Fort, and Aurélie Névéol. Can synthetic text help clinical named entity recognition? a study of electronic health records in French. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2320–2338, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.170. URL `https://aclanthology.org/2023.eacl-main.170`. 90

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. 64, 109

[50] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. BROS: A layout-aware pre-trained language model for understanding documents. *CoRR*, abs/2108.04539, 2021. URL `https://arxiv.org/abs/2108.04539`. 17, 58, 59

[51] Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10767–10775, Jun. 2022. doi: 10.1609/aaai.v36i10.21322. URL `https://ojs.aaai.org/index.php/AAAI/article/view/21322`. 15, 18

[52] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024. 65

[53] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`. xx, 90

[54] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 65–83, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.7. URL `https://aclanthology.org/2020.findings-emnlp.7`. 38

[55] Wen Huang, Hongbin Liu, Minxin Guo, and Neil Gong. Visual hallucinations of multimodal large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9614–9631, Bangkok, Thailand, August 2024. Association for Computational Linguis-

tics. doi: 10.18653/v1/2024.findings-acl.573. URL https://aclanthology.org/2024.findings-acl.573. 6

[56] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022. xx, 18, 21, 24, 47, 59, 63, 65, 90, 91

[57] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sep 2019. doi: 10.1109/icdar.2019.00244. URL http://dx.doi.org/10.1109/ICDAR.2019.00244. 1, 18, 65, 88

[58] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019. 20, 84, 86

[59] ibml. What is document automation? how it works & benefits, 2024. URL https://www.ibml.com/blog/what-is-document-automation-how-it-works-benefits/. v, 1

[60] Global Growth Insights. Document automation software market size, 2024. URL https://www.globalgrowthinsights.com/market-reports/document-automation-software-market-1001. v, 1

[61] Paras Jain, Ajay Jain, Tianjun Zhang, Pieter Abbeel, Joseph Gonzalez, and Ion Stoica. Contrastive code representation learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5954–5971, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.482. URL https://aclanthology.org/2021.emnlp-main.482. 38

[62] Prithwish Jana. Neurosymbolic llm for mathematical reasoning and software engineering. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8492–8493. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/961. URL https://doi.org/10.24963/ijcai.2024/961. Doctoral Consortium. 86

[63] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019. xv, 1, 3, 65, 88, 113, 116, 118

[64] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019. xiii, 7, 16, 18, 53

[65] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating LLM hallucination via self reflection. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore, December 2023. Association for Computational Linguis-

tics. doi: 10.18653/v1/2023.findings-emnlp.123. URL https://aclanthology.org/2023.findings-emnlp.123. 6

[66] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57. 91

[67] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57. 59

[68] Nitish Joshi and He He. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.256. URL https://aclanthology.org/2022.acl-long.256. 46

[69] Dan Jurafsy and James H. Martin. *Speech and language processing*, chapter 23, pages 494–520. 3 edition, 12 2021. 13, 23, 37, 86

[70] Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. Ait-qa: Question answering dataset over complex tables in the airline industry, 2021. 12

[71] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr. 38, 40

[72] Bugeun Kim, Kyung Seo Ki, Donggeon Lee, and Gahgene Gweon. Point to the Expression: Solving Algebraic Word Problems using the Expression-Pointer Transformer Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, November 2020. Association for Computational Linguistics. 12, 30

[73] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 18

[74] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21k.html. 17

[75] Miles A Kimball. Visual design principles: An empirical study of design lore. *Journal of Technical Writing and Communication*, 43(1):3–41, 2013. 6, 14, 59, 60

[76] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 53

[77] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`. 43, 101

[78] David G. Kirkpatrick and John D. Radke. A framework for computational morphology. *Machine Intelligence and Pattern Recognition*, 2:217–248, 1985. URL `https://api.semanticscholar.org/CorpusID:118719120`. 110

[79] David G Kirkpatrick and John D Radke. A framework for computational morphology. In *Machine Intelligence and Pattern Recognition*, volume 2, pages 217–248. Elsevier, 1985. 14

[80] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL `https://aclanthology.org/2024.acl-long.50/`. 92

[81] Michael Krumdick, Rik Koncel-Kedziorski, Viet Dac Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. BizBench: A quantitative reasoning benchmark for business and finance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8309–8332, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.452. URL `https://aclanthology.org/2024.acl-long.452/`. 92

[82] Aviral Kumar and Sunita Sarawagi. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*, 2019. 59

[83] Niraj Kumar. How data lineage is revolutionizing regulatory and compliance, 2023. URL `https://medium.com/@niraj.datametica/how-data-lineage-is-revolutionizing-regulatory-and-compliance-2ef835aced56`. 2

[84] Tomoya Kurosawa and Hitomi Yanaka. Logical inference for counting on semi-structured tables. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 84–96, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-srw.8. URL `https://aclanthology.org/2022.acl-srw.8`. 13

[85] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5), November 2009. ISSN 1550-2376. doi: 10.1103/physreve.80.056117. URL `http://dx.doi.org/10.1103/PhysRevE.80.056117`. 111

[86] VI Lcvenshtcin. Binary coors capable or ‘correcting deletions, insertions, and reversals.

In *Soviet Physics-Doklady*, volume 10, pages 707–710, 1966. 71

[87] Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister. ROPE: Reading order equivariant positional encoding for graph-based document information extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 314–321, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.41. URL `https://aclanthology.org/2021.acl-short.41`. 14, 17, 18, 21, 24, 47, 60, 116, 121

[88] Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. FormNet: Structural encoding beyond sequential modeling in form document information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3735–3754, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.260. URL `https://aclanthology.org/2022.acl-long.260`. 14, 17, 18, 19, 21, 24, 47

[89] Chen-Yu Lee, Chun-Liang Li, Hao Zhang, Timothy Dozat, Vincent Perot, Guolong Su, Xiang Zhang, Kihyuk Sohn, Nikolay Glushnev, Renshen Wang, Joshua Ainslie, Shangbang Long, Siyang Qin, Yasuhisa Fujii, Nan Hua, and Tomas Pfister. FormNetV2: Multimodal graph contrastive learning for form document information extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9011–9026, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.501. URL `https://aclanthology.org/2023.acl-long.501`. xx, 14, 18, 19, 60, 62, 65, 66, 87, 88, 90, 91, 110

[90] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023. 85

[91] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666, 2006. 58, 87, 88

[92] David D. Lewis, Gady Agam, Shlomo Engelson Argamon, Ophir Frieder, David A. Grossman, and Jefferson Heard. Building a test collection for complex document information processing. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006. 54

[93] Moxin Li, Fuli Feng, Hanwang Zhang, Xiangnan He, Fengbin Zhu, and Tat-Seng Chua. Learning to imagine: Integrating counterfactual thinking in neural discrete reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 57–69, Dublin, Ireland, May 2022. Associ-

ation for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.5. URL https://aclanthology.org/2022.acl-long.5. 5, 38, 104

[94] Moxin Li, Fuli Feng, Hanwang Zhang, Xiangnan He, Fengbin Zhu, and Tat-Seng Chua. Learning to imagine: Integrating counterfactual thinking in neural discrete reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 57–69, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.5. URL https://aclanthology.org/2022.acl-long.5. 5, 13, 21, 24, 47

[95] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I. Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, 2021. doi: 10.1109/CVPR46437.2021.00560. 6, 17, 18, 21, 24, 47

[96] Qiwei Li, Zuchao Li, Xiantao Cai, Bo Du, and Hai Zhao. Enhancing visually-rich document understanding via layout structure modeling. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia Workshops*, MMAsia '23 Workshops, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400703263. doi: 10.1145/3611380.3628554. URL https://doi.org/10.1145/3611380.3628554. 18, 65, 66

[97] Yang Li, Yangyang Yu, Haohang Li, Zhi Chen, and Khaldoun Khashanah. Trading-GPT: Multi-Agent System with Layered Memory and Distinct Characters for Enhanced Financial Trading Performance. Papers 2309.03736, arXiv.org, September 2023. URL https://ideas.repec.org/p/arx/papers/2309.03736.html. 92

[98] Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. Compositional generalization for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1438. URL https://aclanthology.org/D19-1438. 13

[99] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1912–1920, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475345. URL https://doi.org/10.1145/3474085.3475345. 58, 59

[100] Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. Neuro-symbolic data generation for math reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 23488–23515. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/29d319f7c1513c9ecd81d3a6e9632a6e-Paper-Conference.pdf. 86

[101] Zhangheng LI, Keen You, Haotian Zhang, Di Feng, Harsh Agrawal, Xiujun Li, Mohana

Prasad Sathya Moorthy, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-UI 2: Mastering universal user interface understanding across platforms. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GBfYgjOfSe. 84, 92

[102] 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, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015. 12

[103] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 83

[104] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 6

[105] Qi Liu, Matt Kusner, and Phil Blunsom. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 187–197, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.18. URL https://aclanthology.org/2021.naacl-main.18. 38

[106] Shuang Liu, Renshen Wang, Michalis Raptis, and Yasuhisa Fujii. Unified line and paragraph detection by graph convolutional networks. In Seiichi Uchida, Elisa Barney, and Véronique Eglin, editors, *Document Analysis Systems*, pages 33–47, Cham, 2022. Springer International Publishing. ISBN 978-3-031-06555-2. 60

[107] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. 25, 27, 31, 39, 43, 61, 101, 102, 109

[108] Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, Chao Li, Sheng Xu, Dezhi Chen, Yun Chen, Zuo Bai, and Liwen Zhang. Fin-r1: A large language model for financial reasoning through reinforcement learning, 2025. URL https://arxiv.org/abs/2503.16252. 92

[109] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. 109

[110] Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. Geolayoutlm: Geometric pretraining for visual information extraction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. xix, 18, 57, 58, 60, 63, 65, 66

[111] Mahmoud Mahfouz, Ethan Callanan, Mathieu Sibue, Antony Papadimitriou, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. The state of the art of large language models on chartered financial analyst exams. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1068–1082, Miami, Florida,

US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.80. URL https://aclanthology.org/2024.emnlp-industry.80/. 92

[112] Arjun Mani, William Hinthorn, Nobline Yoo, and Olga Russakovsky. Point and ask: Incorporating pointing into visual question answering. 2020. 84

[113] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209, 2021. xiv, xix, 1, 7, 18, 19, 71, 72, 73, 76, 88

[114] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 1, 19, 71, 76

[115] McKinsey. Fueling digital operations with analog data, 2022. URL https://www.mckinsey.com/capabilities/operations/our-insights/fueling-digital-operations-with-analog-data. 1

[116] Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur Parikh, and Emma Strubell. Improving compositional generalization with self-training for data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4205–4219, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.289. URL https://aclanthology.org/2022.acl-long.289. 13

[117] Zihang Meng, Licheng Yu, Ning Zhang, Tamara L Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. Connecting what to say with where to look by modeling human attention traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12679–12688, 2021. 84

[118] Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Singh Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. *CoRR*, abs/2204.05660, 2022. URL https://doi.org/10.48550/arXiv.2204.05660. 24

[119] Richard Montague. *The Proper Treatment of Quantification in Ordinary English*, pages 221–242. Springer Netherlands, Dordrecht, 1973. ISBN 978-94-010-2506-5. doi: 10.1007/978-94-010-2506-5_10. URL https://doi.org/10.1007/978-94-010-2506-5_10. 5, 13, 35

[120] Rungsiman Nararatwong, Natthawut Kertkeidkachorn, and Ryutaro Ichise. KIQA: Knowledge-infused question answering model for financial table-text data. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 53–61, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.6. URL https://aclanthology.org/2022.deelio-1.6. 13

[121] Mark Neumann, Zejiang Shen, and Sam Skjonsberg. Pawls: Pdf annotation with labels and structure, 2021. 90

[122] Laura Nguyen, Thomas Scialom, Jacopo Staiano, and Benjamin Piwowarski. Skim-

attention: Learning to focus via document layout. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2413–2427, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.207. URL https://aclanthology.org/2021.findings-emnlp.207. 6

[123] Armineh Nourbakhsh, Cathy Jiao, Sameena Shah, and Carolyn Rosé. Improving compositional generalization for multi-step quantitative reasoning in question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1916–1932. Association for Computational Linguistics, December 2022. URL https://preview.aclanthology.org/emnlp-22-ingestion/2022.emnlp-main.125/. 38, 42

[124] Armineh Nourbakhsh, Sameena Shah, and Carolyn Rose. Towards a new research agenda for multimodal enterprise document understanding: What are we missing? In *Proceedings of the Findings of the 62nd Annual Meeting of the Association for Computational Linguistics*, Bangkok, Thailand, August 2024. Association for Computational Linguistics. 72, 73

[125] Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.225. URL https://aclanthology.org/2020.findings-emnlp.225. 5, 13, 21, 24, 28, 38, 47

[126] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. doi: 10.1609/aaai.v29i1.9602. URL https://ojs.aaai.org/index.php/AAAI/article/view/9602. 78

[127] Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.89. URL https://aclanthology.org/2020.emnlp-main.89. 24

[128] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019. 1, 3, 65, 88

[129] Paycom. The true cost of manual hr processes, 2021. URL https://cdn.paycom.com/mkon/www/media/resources-content/The_True_Cost_of_Manual_Processes.pdf. 1

[130] David Peer, Philemon Schöpf, Volckmar Nebendahl, Alexander Rietzler, and Sebastian Stabinger. Anls*–a universal document processing metric for generative large language models. *arXiv preprint arXiv:2402.03848*, 2024. 19, 73

[131] Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu,

and Haifeng Wang. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *Empirical Methods in Natural Language Processing (Findings)*, 2022. 14, 17, 21, 24, 47

[132] Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, et al. Lmdx: Language model-based document information extraction and localization. *arXiv preprint arXiv:2309.10952*, 2023. 84, 92

[133] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3743–3751, 2022. 88

[134] Karmvir Singh Phogat, Sai Akhil Puranam, Sridhar Dasaratha, Chetan Harsha, and Shashishekar Ramakrishna. Fine-tuning smaller language models for question answering over financial documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10528–10548, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.617. URL https://aclanthology.org/2024.findings-emnlp.617/. 92

[135] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020. 8, 84

[136] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 732–747. Springer, 2021. 17, 18, 21, 24, 47

[137] Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. Grounding language model with chunking-free in-context retrieval. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1311, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.71. URL https://aclanthology.org/2024.acl-long.71. 73

[138] Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Han Yi, Jimin Huang, Qianqian Xie, and Jianyun Nie. Fino1: On the transferability of reasoning enhanced llms to finance, 2025. URL https://arxiv.org/abs/2502.08127. 92

[139] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. GraphIE: A graph-based framework for information extraction. 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 751–761, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1082. URL https://aclanthology.org/N19-1082. 16

[140] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`. 86, 92

[141] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 17

[142] 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. 17

[143] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=HPuSIXJaa9`. 92

[144] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`. 15, 50

[145] Natraj Raman, Sameena Shah, and Manuela Veloso. Synthetic document generator for annotation-free layout recognition. *CoRR*, abs/2111.06016. URL `https://arxiv.org/abs/2111.06016`. 6, 51

[146] Natraj Raman, Sameena Shah, and Manuela Veloso. Synthetic document generator for annotation-free layout recognition. *Pattern Recognition*, 128:108660, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.108660. URL `https://www.sciencedirect.com/science/article/pii/S0031320322001418`. 90

[147] Pritika Ramu, Sijia Wang, Lalla Mouatadid, Joy Rimchala, and Lifu Huang. re^2: Region-aware relation extraction from visually rich documents. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8731–8747, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.484. URL `https://aclanthology.org/2024.naacl-long.484`. 60

[148] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:

10.18653/v1/D19-1251. URL https://aclanthology.org/D19-1251. 4, 13, 21, 24, 47

[149] Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1033. URL https://aclanthology.org/K19-1033. 4, 13, 20, 24, 25

[150] Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. DocFinQA: A long-context financial reasoning dataset. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 445–458, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.42. URL https://aclanthology.org/2024.acl-short.42/. 92

[151] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 6, 17

[152] Adithya Renduchintala and Adina Williams. Investigating failures of automatic translationin the case of unambiguous gender. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3454–3469, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.243. URL https://aclanthology.org/2022.acl-long.243. 91

[153] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 17, 18

[154] Subhro Roy, Tim Vieira, and Dan Roth. Reasoning about Quantities in Natural Language. *Transactions of the Association for Computational Linguistics*, 3:1–13, 01 2015. ISSN 2307-387X. doi: 10.1162/tacl_a_00118. URL https://doi.org/10.1162/tacl_a_00118. 12, 24

[155] Gaurav Sahu, Abhay Puri, Juan A. Rodriguez, Amirhossein Abaskohi, Mohammad Chegini, Alexandre Drouin, Perouz Taslakian, Valentina Zantedeschi, Alexandre Lacoste, David Vazquez, Nicolas Chapados, Christopher Pal, Sai Rajeswar, and Issam H. Laradji. Insightbench: Evaluating business analytics agents through multi-step insight generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZGqd0cbBvm. 92

[156] Raeid Saqur and Karthik Narasimhan. Multimodal graph networks for compositional generalization in visual question answering. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural*

Information Processing Systems, volume 33, pages 3070–3081. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1fd6c4e41e2c6a6b092eb13ee72bce95-Paper.pdf. 13

[157] Imanol Schlag, Paul Smolensky, Roland Fernandez, Nebojsa Jojic, Jürgen Schmidhuber, and Jianfeng Gao. Enhancing the transformer with explicit relational encoding for math problem solving. *arXiv preprint arXiv:1910.06611*, 2019. 19

[158] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://aclanthology.org/P17-1099. 60, 63

[159] Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G. Parameswaran, and Eugene Wu. Docetl: Agentic query rewriting and evaluation for complex document processing, 2025. URL https://arxiv.org/abs/2410.12189. 92

[160] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300. 86, 92

[161] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074. 14, 50

[162] Štěpán Šimsa, Milan Šulc, Michal Uřičář, Yash Patel, Ahmed Hamdi, Matěj Kocián, Matyáš Skalickỳ, Jiří Matas, Antoine Doucet, Mickaël Coustaty, et al. Docile benchmark for document information localization and extraction. *arXiv preprint arXiv:2302.05658*, 2023. 3, 88

[163] Georgios Spithourakis, Isabelle Augenstein, and Sebastian Riedel. Numerically grounded language models for semantic error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 987–992, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1101. URL https://aclanthology.org/D16-1101. 4

[164] Georgios Spithourakis, Steffen Petersen, and Sebastian Riedel. Clinical text prediction with numerically grounded conditional language models. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 6–16, Auxtin, TX, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-6102. URL https://aclanthology.org/W16-6102. 4

[165] Pragya Srivastava, Manuj Malik, Vivek Gupta, Tanuja Ganu, and Dan Roth. Evaluating LLMs' mathematical reasoning in financial document question answering. In Lun-Wei

Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3853–3878, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.231. URL https://aclanthology.org/2024.findings-acl.231/. 92

[166] Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, pages 564–579. Springer, 2021. 1, 87, 88

[167] Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: Key information extraction datasets involving long documents with complex layouts. *arXiv preprint arXiv:2105.05796*, 2021. 18

[168] Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. Towards table-to-text generation with numerical reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1451–1465, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.115. URL https://aclanthology.org/2021.acl-long.115. 4, 24

[169] Kexuan Sun, Harsha Rayudu, and Jay Pujara. A hybrid probabilistic approach for table understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5): 4366–4374, May 2021. doi: 10.1609/aaai.v35i5.16562. URL https://ojs.aaai.org/index.php/AAAI/article/view/16562. 21, 24, 47

[170] Stacey Svetlichnaya. Deepform: Understand structured documents at scale, 2020. URL https://wandb.ai/stacey/deepform_v1/reports/DeepForm-Understand-Structured-Documents-at-Scale--VmlldzoyODQ3Njg. 87, 88

[171] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264, 2023. 17, 18, 19, 59, 65, 90

[172] Avijit Thawani, Jay Pujara, and Filip Ilievski. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6960–6967, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.557. URL https://aclanthology.org/2021.emnlp-main.557. 4, 12

[173] Avijit Thawani, Jay Pujara, Pedro Szekely, and Filip Ilievski. Representing numbers in nlp: a survey and a vision. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021. 4, 12

[174] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation.

In *International Conference on Learning Representations*, 2020. 64

[175] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Document collection visual question answering. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 778–792. Springer, 2021. 19, 71

[176] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109834. URL `https://www.sciencedirect.com/science/article/pii/S0031320323005320`. 19, 71, 76

[177] Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 7821–7846. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/0ef1afa0daa888d695dcd5e9513bafa3-Paper-Conference.pdf`. 86

[178] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019. 110

[179] Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19528–19540, 2023. 1, 19, 71, 76, 88

[180] Gaël Varoquaux, Alexandra Sasha Luccioni, and Meredith Whittaker. Hype, sustainability, and the price of the bigger-is-better paradigm in ai, 2025. URL `https://arxiv.org/abs/2409.14160`. 83

[181] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models, 2024. URL `https://arxiv.org/abs/2412.13303`. 84

[182] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`. 5, 27

[183] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`. 14, 15, 17

[184] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 50

[185] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`. 112

[186] Gaurav Verma, Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Tucker Balch, and Manuela Veloso. Adaptagent: Adapting multimodal web agents with few-shot learning from human demonstrations, 2024. URL `https://arxiv.org/abs/2411.13451`. 92

[187] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf`. 100

[188] Dongsheng Wang, Zhiqiang Ma, Armineh Nourbakhsh, Kang Gu, and Sameena Shah. Docgraphlm: Documental graph language model for information extraction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1944–1948, 2023. 18

[189] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv:2401.00908*, 2023. 15, 18, 59, 65, 90

[190] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6, 18, 77, 80

[191] R. Wang, Y. Fujii, and A. C. Popat. Post-ocr paragraph recognition by graph convolutional networks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2533–2542, Los Alamitos, CA, USA, jan 2022. IEEE Computer Society. doi: 10.1109/WACV51458.2022.00259. URL `https://doi.ieeecomputersociety.org/10.1109/WACV51458.2022.00259`. 16, 17

[192] R. Wang, Y. Fujii, and A. C. Popat. Post-ocr paragraph recognition by graph convolutional networks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2533–2542, Los Alamitos, CA, USA, jan 2022. IEEE Computer Society. doi: 10.1109/WACV51458.2022.00259. URL `https://doi.ieeecomputersociety.org/10.1109/WACV51458.2022.00259`. 60

[193] Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Pro-*

cessing, pages 845–854, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1088. URL https://aclanthology.org/D17-1088. 4

[194] Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1088. URL https://aclanthology.org/D17-1088. 13

[195] Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193, 2023. 88

[196] William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. LAW: Legal agentic workflows for custody and fund services contracts. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, Steven Schockaert, Kareem Darwish, and Apoorv Agarwal, editors, *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 583–594, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-industry.50/. 92

[197] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J. 5

[198] Jialin Wu, Xia Hu, Yaqing Wang, Bo Pang, and Radu Soricut. Omni-smola: Boosting generalist multimodal models with soft mixture of low-rank experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14205–14215, 2024. 18

[199] Jialin Wu, Xia Hu, Yaqing Wang, Bo Pang, and Radu Soricut. Omni-smola: Boosting generalist multimodal models with soft mixture of low-rank experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14205–14215, 2024. 77, 90

[200] Qinzhuo Wu, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. Math word problem solving with explicit numerical values. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5859–5869, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.455. URL https://aclanthology.org/2021.acl-long.455. 12, 13

[201] Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xinrun Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Guanglin Niu, Tongliang Li, and Zhoujun Li. Tablebench: A comprehensive and complex benchmark for table question answer-

ing, 2025. URL `https://arxiv.org/abs/2408.09174`. 92

[202] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023. 92

[203] Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi, Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. *arXiv preprint arXiv:2406.11633*, 2024. 84

[204] Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, Bo Li, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 2: Text. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=LWD7upg1ob`. 84

[205] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2579–2591, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.201. URL `https://aclanthology.org/2021.acl-long.201`. 6, 15, 17, 18, 21, 24, 47, 48, 50, 54

[206] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.201. URL `https://aclanthology.org/2021.acl-long.201`. 59, 65

[207] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1192–1200, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403172. URL `https://doi.org/10.1145/3394486.3403172`. 17, 18, 19, 21, 24, 47, 50, 59

[208] Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. Multi-scale contrastive knowledge co-distillation for event temporal relation extraction, 2024. 64

[209] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*, 2023. 65, 90

[210] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. xx, 90

[211] Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Annual Conference of the Association for Computational Linguistics (ACL)*, July 2020. 13, 21, 24, 47

[212] Pengcheng Yin, Hao Fang, Graham Neubig, Adam Pauls, Emmanouil Antonios Platanios, Yu Su, Sam Thomson, and Jacob Andreas. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.225. URL `https://aclanthology.org/2021.naacl-main.225`. 5, 13, 21, 24, 27, 28, 37, 47

[213] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 84, 92

[214] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR, 2018. 59, 63

[215] Chongjian Yue, Xinrun Xu, Xiaojun Ma, Lun Du, Hengyu Liu, Zhiming Ding, Yanbing Jiang, Shi Han, and Dongmei Zhang. Enabling and analyzing how to efficiently extract information from hybrid long documents with llms, 2024. URL `https://arxiv.org/abs/2305.16344`. 92

[216] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007. doi: 10.1109/TPAMI.2007.1078. 40

[217] Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. Reading order matters: Information extraction from visually-rich documents by token path prediction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13716–13730, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.846. URL `https://aclanthology.org/2023.emnlp-main.846`. 62, 87

[218] Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. Graph-to-tree learning for solving math word problems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3928–3937, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.362.

URL https://aclanthology.org/2020.acl-main.362. 4

[219] Le Zhang, Zichao Yang, and Diyi Yang. TreeMix: Compositional constituency-based data augmentation for natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5243–5258, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.385. URL https://aclanthology.org/2022.naacl-main.385. 38

[220] Qiyuan Zhang, Lei Wang, Sicheng Yu, Shuohang Wang, Yang Wang, Jing Jiang, and Ee-Peng Lim. NOAHQA: Numerical reasoning with interpretable graph question answering dataset. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4147–4161, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.350. URL https://aclanthology.org/2021.findings-emnlp.350. 8, 13, 21, 24, 47

[221] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. Do language embeddings capture scales? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4889–4896, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.439. URL https://aclanthology.org/2020.findings-emnlp.439. 4

[222] Yue Zhang, Zhang Bo, Rui Wang, Junjie Cao, Chen Li, and Zuyi Bao. Entity relation extraction as dependency parsing in visually rich documents. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2759–2768, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.218. URL https://aclanthology.org/2021.emnlp-main.218. 57

[223] Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. LONGAGENT: Achieving question answering for 128k-token-long documents through multi-agent collaboration. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16310–16324, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.912. URL https://aclanthology.org/2024.emnlp-main.912/. 92

[224] Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.454. URL https://aclanthology.org/2022.acl-long.454. 1, 4, 5, 29, 42

[225] Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In

Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.852. URL `https://aclanthology.org/2024.acl-long.852/`. 92

[226] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 697–706, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society. doi: 10.1109/WACV48630.2021.00074. URL `https://doi.ieeecomputersociety.org/10.1109/WACV48630.2021.00074`. 29

[227] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL `https://aclanthology.org/2021.acl-long.254`. 1, 4, 5, 12, 29, 30, 35, 38, 42, 100

[228] Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat-Seng Chua. Tat-llm: A specialized language model for discrete reasoning over tabular and textual data, 2024. 1

[229] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded Question Answering in Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[230] Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1161. URL `https://aclanthology.org/P19-1161`. 38

[231] Ran Zmigrod, Zhiqiang Ma, Armineh Nourbakhsh, and Sameena Shah. Treeform: End-to-end annotation and evaluation for form document parsing. *arXiv preprint arXiv:2402.05282*, 2024. 73

[232] Ran Zmigrod, Pranav Shetty, Mathieu Sibue, Zhiqiang Ma, Armineh Nourbakhsh, Xiaomo Liu, and Manuela Veloso. "what is the value of templates?" rethinking document information extraction datasets for LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13162–13185, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-emnlp.770`. 6, 73

[233] Ran Zmigrod, Dongsheng Wang, Mathieu Sibue, Yulong Pei, Petr Babkin, Ivan Brugere,

Xiaomo Liu, Nacho Navarro, Antony Papadimitriou, William Watson, Zhiqiang Ma, Armineh Nourbakhsh, and Sameena Shah. Buddie: A business document dataset for multi-task information extraction, 2024. 3, 65, 88