Understanding How Vision-Language Models *Reason* when Solving Visual Math Problems

Joseph Tey
Stanford,CA
joetey@stanford.edu

Abstract

Frontier vision language models (VLMs) struggle to solve visual math problems. Inspired by LLM reasoning enhancements, one promising approach to improve this skill is to fine-tune VLMs on intermediate, chain-of-thought reasoning samples. While several studies have explored the efficacy of various types of reasoning data, little effort exists to deeply understand, on an attention-level, why such techniques work. In fact, a recent study found that most of these VLMs actually struggle to understand the visual component of these problems, and instead, rely heavily on textual cues instead. Motivated by this, we propose a novel image-to-text attention ratio that quantifies the extent to which a model relies on visual-only cues, as a potential indicator for effective reasoning for visual math problems. We fine-tune a pre-trained VLM on various reasoning samples, and conduct an extensive quantitative and qualitative analysis of the models' reasoning capabilities using the ratio. Not only does fine-tuning VLMs on reasoning samples improve the accuracy of these models, but such results are aligned with their image-to-text attention ratios, showing promise that a high ratio may indicate greater elicitation of useful, vision-only reasoning segments. Qualitative analysis of token heat-maps also reveal that areas with a high image-to-text attention ratio are indeed, as hypothesized, often correlated with visual-only information.

1. Introduction

With the advent of large language models, math problems have always challenged such mechanisms. This is no exception for Vision Language Models (VLMs), a new architectural family of models that seemingly *understand* both image and textual data together (in relation to one another). Frontier VLMs struggle with *visual math reasoning*, and of all performance categories evaluated in the recent Phi-3-Vision technical report [1], visual math reasoning spans the lowest performance between 20% and 50%, across all compared models. Of course, efforts to replicate 'frontier' performance have been substantive (albeit still a gap), with notable models including LLaVA [14], Mini-GPT [25], and Intern-VL [6].

Visual math reasoning includes the ability to effectively reason and solve mathematical problems that have an explicit visual component. For example, geometry problems often rely on shapes, lines, angles, and functions and graphs require an understanding of how numbers appear on some x-y plane. Improving a VLM's ability to understand such problems has huge potential in real-world fields, particularly in education. Students are able to engage in interactive conversations with a model that is able to *see* what they're looking at; whether it is a problem, or their own diagrams.

Inspired by LLM enhancement techniques, one promising approach to improve this skill is to fine-tune open source VLMs on explicit visual reasoning data related visual math problems, or what scholars call 'rationales' [12] [22]. Often, such data is generated synthetically with a frontier model (e.g. GPT-o, Claude), before being distilled to a smaller model.

While there have been many studies exploring different types of synthetic data to improve this capability across VLMs, including agentic rationales [18], problem decomposition [23], multi-class techniques [9], and more, few have attempted to deeply understand *why* certain techniques work, and importantly, compare different knowledge distillation approach through an interpret-ability lens.

In fact, a recent study by Zhang [21] found that most of these fine-tuned VLMs struggle to actually understand the *visual* component of these problems, and instead, rely heavily on textual cues instead. As such, this paper seeks to understand: For solving visual math problems, what makes a good *rationale*? How can we understand a model's *reasoning* capabilities through different heuristics and indicators?

We hope this not only helps researchers understand what specific sort of reasoning VLMs should try and elicit, helping them craft better synthetic data, but also for us to learn about these models think and reason underneath it all.

Our contributions are primarily two-fold:

- Our hypothesis, primarily motivated by Zhang's work [21], is that effective reasoning should rely equally, if not more, on visual elements of the diagram. As such, we propose a novel metric to quantify the degree to which a VLM's reasoning and prediction attends to the image, compared to other tokens, called the *image-to-text* attention ratio. Our belief is that this ratio is indicative of a model's ability to elicit vision-only properties of a visual math problem.
- To better understand this metric, we fine-tune a TinyLLaVA3.1b model using various rationales, ranging across different levels of complexity, and quantitatively and qualitatively evaluated their outputted reasoning using our approach to try and understand what these models 'attend' to. We focus on geometry problems, and evaluate these models on the Inter-GPS dataset (601 pairs in test-set, 2401 in training set).

2. Related Work

2.1. Vision Language Models

The impressive performance of GPT-4o, GPT-V, as well as Claude vision models has shocked multi-modal research communities. Upon their release, these models significantly outperformed open-source models of the time, sparking wide research efforts to devise new strategies to improve the performance of these smaller models. These efforts have targeted a multifaceted range of model abilities, including captioning (can these models comprehensively describe what they see?), spatial intelligence (can these models accurately pin point specific regions on images?), reasoning (can these models go beyond just describing, but problem-solving based on what they see?), and many more. All of these abilities have important applications in many fields; for example, building web-based agents requires multi-modal models to accurately identify precise regions on images in order to interact with them.

One successful approach to bridge this gap between larger models and smaller models is visual instruction tuning [14], as first demonstrated by models such as LLaVa and Mini-GPT [25]. This technique involves freezing the visual encoder as well as the language model, pre-training in order to align these two components via a projection layer, before fine-tuning using high-quality, synthetically generated instruction data that cover a range of different tasks. Different models are better at different types of tasks, depending on how the model was fine-tuned.

2.2. Improving Reasoning in LLMs

Attempts to improve reasoning in VLMs have been largely inspired by strides made for traditional LLMs. We describe two traditional techniques in this section.

A popular approach to improving reasoning in large language models is chain-of-thought reasoning [19], directly prompting these models with *"Let's think step-by-step"* in order for these models to output their step-by-step thinking process, before outputting the answer. This has shown to improve LLM performance considerably.

Some have used this technique to improve smaller models by fine-tuning them on *reasoning* samples or rationales; that is, intermediate, chain-of-thought reasoning data synthetically generated by larger, more powerful models [8]. This gives researchers more control by allowing them to explicitly craft different types of reasoning data that will nudge LLMs to reason in a particular way, while also leveraging the superior reasoning capabilities of frontier models. It is hypothesized that such fine-tuning elicits a smaller model's underlying ability to also reason in similar ways.

We leverage both of these techniques to craft various reasoning, chain-of-thought samples using frontier models, before fine-tuning smaller models with such data, before conducting an extensive analysis on such reasoning.

2.3. Existing Techniques to Improve Visual Math Reasoning in VLMs

Zhang [22] used human-annotated reasoning samples extracted from lectures and custom-made explanations, in order to try and enhance reasoning. However, these reasoning samples are not necessarily related to the image; they are just generic 'explanations', and not necessarily specific to the visual diagram. Jia [12] uses a larger model to generate a detailed description of the image, before using this description to try and generate the reasoning sample. However, a 'gold standard' reasoning sample was synthetically generated without the MLM; also again, the reasoning is not directly related to *visual* reasoning. Lu's [16] few-shot GPT-3.5 model is non-multi-modal, and does not include visual features entirely as well. Other approaches have used interesting *reasoning* samples such as decomposing the problem into smaller sub-problems as some chain-of-thought equivalence [23]. Some have even used micro agent architectures in order to 'plan' a strategy to solve the problem [18], and using this as a sample to fine-tune smaller models to adopt such *agent* behavior in how they normally reason and think.

2.4. Interpreting and Understanding VLMs

There is a rich literature in understanding the decision-making systems of vision models on an attention-level. One of the main challenges in using attention weights as an explainable, interpretable heuristic is the challenge of effectively aggregating the weights of multiple layers and attention heads [5]. Techniques such as attention roll-out or attention flow [2] have been used to account for residual or skip connections in quantifying their *flow* across different layers. However, such techniques, given how vision models

(e.g. ViT, CLIP, etc.) are encoders, are mostly designed for self-attention mechanisms.

As VLMs have the novel property of ingesting both text and image data, while outputting text, this study focuses primarily on interpreting encoder-to-decoder attention weights. We seek to understand how a model's outputted reasoning reflects what the model is 'attending' to. Given the recency of such research, there have been few efforts in to build better, more insightful interpret-ability tools for multi-modal models that ingest both image and text data. Recently, Intel released LVLM-Interpret [17], a tool for better understanding multi-modal vision language models. At the time of writing this paper, the team had not yet released their tool for public experimentation. Yet, the paper seems impressive, building upon a core encoder-decoder attention flow mechanism pioneered by Chefer [4]. Few have employed attention-level explainable techniques for understanding VLM performance on visual math reasoning specifically, which is the primary focus of this paper.

3. Dataset

We will be using the Inter-GPS [15] dataset to fine-tune the models, as well as evaluate their performance. This dataset has $2,401$ QA pairs in the training split, as well as 601 QA pairs in the test split. This data-set was chosen because unlike many other math evaluation datasets, Inter-GPS contains a substantive training dataset that will be helpful in this study. This dataset is specific to **geometry problems**, which will be the specific type of visual math problem this study focuses on. Each entry in this data-set contains a **Problem Text**, a **Diagram** (image), **Choices**, and **Diagram Literals** (textual description of the diagram). See Figure 1 for an example.

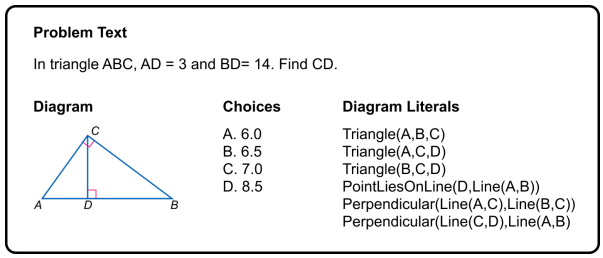

Figure 1. Example of a problem in the Inter-GPS dataset

4. Methods

To analyze a VLM's reasoning for solving visual math problems, we employ a four-stage approach.

1. **Synthetic Data Generation:** For each of the $2,401$ geometry problems in the training split of the dataset, we use GPT-4o to generate 3 different types of intermediate reasoning samples.

2. **Model Preparation & Fine-Tuning:** We prepare variants of TinyLLaVA3.1B, some of which are just pre-trained, and others which are fine-tuned using the generated synthetic data.

3. **Evaluation:** We evaluate the performance of the prepared models in the previous stage on 601 geometry problems in the test set, and while doing so, save the attention weights for analysis in the next stage.

4. **Interpret-ability Analysis:** We calculate an average *image-to-text* attention ratio for each model variant to analyze what each type of reasoning attends to, and compare this across models. We also do an in-depth analysis of each reasoning sample, token-by-token to better understand what this ratio truly means.

4.1. Synthetic Data Generation

We start by preparing the reasoning data that we will use to fine-tune smaller models. There are various approaches to how a 'reasoning sample' should be structured. Some have explored concatenating the rationale followed by the answer, some concatenate the answer followed by the rationale, and others have employed prefix-tuning techniques in a multi-task framework [9] where the rationale and the answer have separate loss functions. We follow Wei [19], and chose the first type of reasoning sample: rationale, followed by the answer. We explore 3 different types of rationales, which are depicted in Figure 2.

Visual CoT. While various types of complex reasoning samples have been tried, surprisingly, there has been little effort to construct rationales that directly reason from the diagram itself. Other approaches are more grounded in visual descriptions, problem de-compositions, agentic workflows, but what if we just extracted a frontier model's direct ability to reason, in a step-by-step manner, and solve a visual math problem?

Visual CoT + Diagram Literals. Inspired by Jia[12], who generated textual descriptions of their diagrams to enhance rationale generation in the *Describe-Then-Reason* approach, we pre-pended the diagram literal data included in the Inter-GPS to the existing Visual CoT rationale. This is hypothesized to give extra context to the model to understand the visual elements of the diagram.

Visual CoT + Symbolic Solver. Often, with math problems, there are two types of information in the rationale: reasoning and computation. LLMs are known to be poor as the latter (quirks with the tokenizer, etc.), and so some have combined LLMs with symbolic solvers with pre-defined formal notation and rules that have been shown to improve math computation abilities. For this reasoning sample, we follow the prompting rules of the Peano symbolic solver [7].

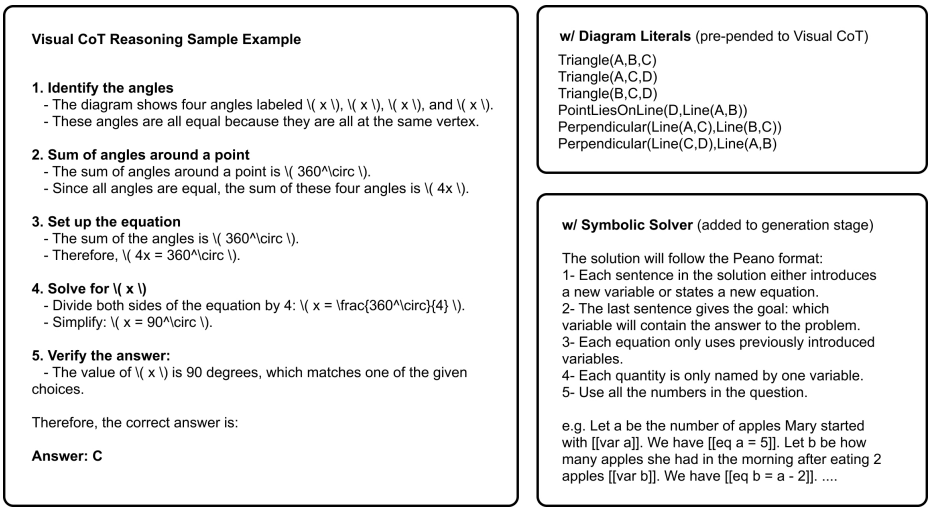

Figure 2. Examples of different reasoning samples generated by GPT-4o

We also choose GPT-4o as the frontier model to generate reasoning samples due to its SOTA vision capabilities, especially in visual math reasoning, as well as its significantly reduced cost.

Let D_{train} be our training dataset of 2401 problems, where $D = \{(p_i, d_i, c_i, a_i)\}_{i=1}^{2401}$, where p is the problem text, d is the diagram image, c is the choices, a is the correct answer. We generate $R_{\text{Visual CoT}}$, $R_{\text{w/ Diagram Literals}}$, and $R_{\text{w/ Symbolic Solver}}$, where R is a set of synthetically generated reasoning samples based on D. Formally, $R = \{r_j\}_{j=1}^{2401}$ and r_j is the j_{th} reasoning sample for each of the 3 categories. To generate each reasoning sample:

$$r_j = \text{GPT}(p_j, d_j, c_j, a_j)$$

where GPT refers to an API call to OpenAI via a specific prompt. See Figure 3 for a sample prompt used to generate a **Visual CoT** reasoning sample.

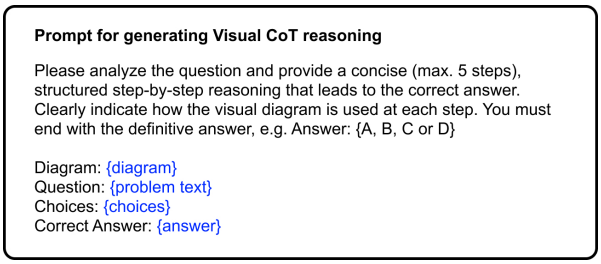

Figure 3. Prompt used to generate Visual CoT reasoning sample using GPT-4o

4.2. Model Preparation & Fine-Tuning

TinyLLaVA [24] was used as our base model for all of our experiments. We specifically load the checkpoints for the TinyLLaVA3.1b model from hugging face, which is their best performing model. We chose this model due to its comparable performance with existing 7B models such as LLaVA-1.5 [13] and Qwen-VL [3], modularizable codebase [11], while at the same time, being only 3.1b parameters (compute was limited).

While we refer to the original paper for more details [24], the architecture of TinyLLaVA consists of a small-scale LLM F_θ, which we opt to use Phi-2 (2.7B), a vision encoder V_γ, which we opt to use SigLIP (0.4B) [20], as well as a connector P_ϕ.

To fine-tune this model, we use LoRA (low-rank adaptation) [10] to reduce the number of parameters (θ' and γ' are the learnable parameters) that we need to fine-tune (rank $= 32$, $\alpha = 64$), with a learning rate of $2e-5$ over 3 epochs. As per Zhou's [24] description, TinyLLaVA maximizes the log-likelihood of the reasoning samples autoregressively as the training objective.

$$\max_{\phi,\gamma',\phi'} \sum_{i=1}^{N} \log F_\theta(r_i | P_{\phi'} \circ V_{\gamma'}(d_i))$$

where N is the length of the reasoning sample, and d_i is the corresponding image.

Using these techniques, we prepare 5 different model variants (with different reasoning samples) to evaluate and analyze:

1. **Vanilla Direct:** Only pre-trained TinyLLaVA3.1b, prompted to directly output the answer (A, B, C or D)
2. **Vanilla CoT:** Only pre-trained TinyLLaVA3.1b, prompted to reason step-by-step (0-shot).
3. **Fine-Tuned Visual CoT:** Used $R_{\text{Visual CoT}}$ to fine-tune TinyLLaVA3.1b

4. **Fine-Tuned Visual CoT + Diagram Literals:** Used $R_{\text{w/ Diagram Literals}}$ to fine-tune TinyLLaVA3.1b

5. **Fine-Tuned Visual CoT + Symbolic Solver:** Used $R_{\text{w/ Symbolic Solver}}$ to fine-tune TinyLLaVA3.1b

4.3. Evaluation

Finally, we evaluate each one of the 5 prepared models on D_{test} by running inference for the 601 test items. We extract the raw answer (A, B, C, and D) using GPT-4o in order to calculate the accuracy against the true class. For every test item i we evaluate each model on, we also output the raw attention weights to analyze in the next stage. See Appendix 9.1 for the inference prompts we used when evalauting these models.

4.4. Interpretability Analysis

Our primary hypothesis is that effective reasoning should rely equally, if not more, on visual elements of the diagram, compared to the problem text and/or choices in the input tokens. As such, for a model's specific output o_i for problem i (o_i includes both the reasoning and the answer), we calculate a novel *image-to-text* attention ratio in an attempt to quantify the extent to which a VLM's outputted reasoning actually *attends* to the image patch tokens, compared to the input text tokens. We wonder if this ratio could be an indicator for *good* reasoning, or perhaps more precisely, a heuristic for a model's ability to deeply extract information accessible only in the visual diagram.

4.4.1 Image-To-Text Attention Ratio

So, how do we calculate the *image-to-text* attention ratio? Firstly, for simplicity, as we have 32 decoder layers and 32 attention heads, we build an average attention map across all layers and heads denoted A^i for problem i. Since A^i is a square, lower triangular matrix, $A^i_{r,c}$ corresponds to the degree to which token r attends to token c (attention weight).

We let $o_i = [t_1, t_2, ..., t_n]$, where o_i is a particular VLM's output for problem i, and n is the number of tokens in the output. For each output token t_j, where $j = 1, 2, ..., n$, we want to calculate the magnitude in which this output token attends to the image patch tokens, compared to the input text tokens. As such, using A_i, we find the average of the attention weights of all the tokens that are associated to the visual patches, denoted V_j, and do the same for the problem text tokens, denoted P_j. Since our SigLIP encoder produces 729 image visual patch tokens, which is always more than the number of problem text tokens, it is important to find the *average* so that reasoning length does not influence this metric. Finally, we calculate $\frac{V_j}{P_j}$ to find the image-to-text attention ratio for token t_j. Hence, to calculate the *image-to-text* attention ratio Q_i for model output o_i

$$Q_i = \sum_{j=1}^{n} (\frac{V_j}{P_j})$$

where n is the number of tokens in the output, V_j is the average attention weight of token t_j across the 729 visual image tokens, and P_j is the average attention weight of token t_j across a variable number of problem text tokens.

Given some output o_i for problem i, this ratio is designed to reflect on average, is the model *reasoning* about things that are *exclusive* (high V_j, low P_j) to the visual patches on a *deeper* or more *consistent* level?

5. Results

In this section, we aim to answer two main questions:

1. Does fine-tuning TinyLLAVA3.1b on direct, *visual* chain-of-thought reasoning samples improve accuracy, compared to our baselines?

2. Is there any connection between the average image-to-text attention ratios of these models' predictions, with their performance accuracy?

Looking at Table 1, our frontier model GPT-4o used for the synthetic data generation stage has a relatively high accuracy of **55.91%**, which is the current SOTA, and outperforms other frontier VLMs like Phi-3, GPT-4-Turbo and Gemini 1.5 Pro. This is promising, as the reasoning samples generated by our 'teacher' model seem to be fairly accurate. Ho [8] found that the better a teacher's reasoning, the better of a 'teacher' they are to the smaller models. Due to cost limitations, we were unable to conduct a more extensive evaluation process of these reasoning samples, such as the multi-CoT evaluation strategy employed by Zhang [21], but this seemed sufficient for this study.

Interestingly, fine-tuning TinyLLaVA3.1b on just the Visual CoT data has a sizeable improvement of **6.33%** over zero-shot CoT on the pre-trained model. In qualitatively analyzing dozens of model outputs, we hypothesize that this is due to the structured, step-by-step reasoning style (see figure 2) induced by visual CoT reasoning data. The chain-of-thought process is quite standard; the model starts by identifying any crucial information in the problem text or in the diagram, before applying relevant theorems, and crunching numbers. This structured process also, in some sense, clearly distinguishes between *reasoning* and *computation*, where the headlines of each step are clearly higher-level reasoning steps, and computation exists in the sub-text. By adopting this frame-work of some sort, it seems as though the VLM is able to apply this general framework to solve unseen problems. This behavior is different in the **Vanilla CoT** model, which has much less structured reasoning.

Model	Accuracy (%)	Image-to-Text Attention Ratio
Vanilla Direct	15.97	0.20
Vanilla CoT	20.63	0.80
Fine-Tuned Visual CoT	26.96	0.93
Fine-Tuned Visual CoT + Diagram Literals	**28.95**	**1.14**
Fine-Tuned Visual CoT + Symbolic Solver	23.46	0.93
GPT-4o CoT	**55.91**	n/a

Table 1. Accuracy and image-to-text attention ratio of model variants on 601 test pairs in Inter-GPS dataset. Ratio cannot be calculated for GPT-4o as we do not have access to the attention weights.

Adding diagram literals also seems to improve accuracy marginally (around **2%**). This model, however, only *sometimes* outputs the diagram literals it was fine-tuned on (and if it does, it often infinitely repeats itself), perhaps indicating that it struggles to generate textual descriptions of the diagram. However, we hypothesize that adding this data in the reasoning samples may enhance the model's general vision comprehension capabilities. The symbolic solver seemed to confuse the model, more-so than it did help it, which may indicate that perhaps the lackluster reasoning capabilities of smaller LLMs (Phi-2) are unable to adhere to the formal, stringent rules of symbolic solvers (see figure 2), in a way that frontier LLMs are able to.

Overall, however, to answer Question 1, fine-tuning TinyLLaVA3.1b on reasoning samples clearly shows improvements upon pre-trained direct and CoT baselines.

Table 1 also shows that a higher image-to-text attention ratio also loosely (not statistically rigorous) correlates to higher accuracy. **Fine-Tuned Visual CoT + Diagram Literals** has the highest image-to-text attention ratio at **1.14**, and also has the highest accuracy of **28.95%**. Likewise, the **Fine-Tuned Visual CoT** model has a 0.13 increase in image-to-text attention ratio compared to the **Vanilla CoT** model. This is promising in hinting that it is the models who are able to more explicitly attend to vision-only information, and perhaps maintain a greater connection to these image tokens as they *reason*, that perform better.

However, this claim is still limited, as there are various forces that create a 'good reasoning sample'; only one of which, may be the extent to which it focuses on the visual diagram. Instead, in the next section, we shift out focus to understanding this metric on a deeper level, on a token-to-token level, trying to visualize what this image-to-text attention ratio actually correlates with.

That is, we seek to answer the question: if a model has a high image-to-text attention ratio, what does this exactly mean on a token level?

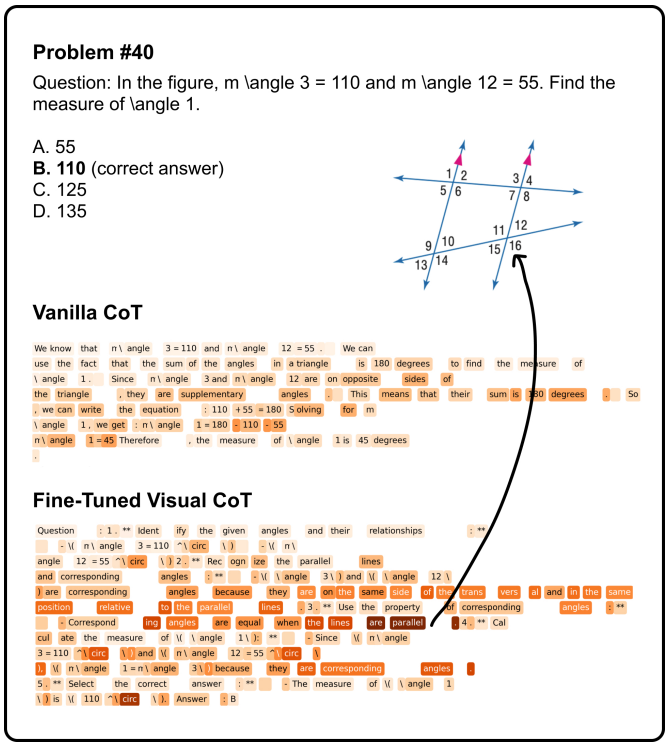

Figure 4. Token heat-map of the *image-to-text* attention ratio for problem 40, where the darker the colour of the token, the higher the ratio, and vice versa.

6. Discussion

In this section, we aim to better understand, through a qualitative lens, what *exactly* the *image-to-text* attention ratio tells us, as well as some of the assumptions/limitations of this ratio as an explainable metric. We hope that learning about this will help us understand what good visual math reasoning looks like, and hopefully, provide more insights as to how to craft *better* reasoning samples to improve the visual reasoning skills of VLMs.

To provide more granular details, figure 4 shows a token heat-map for the outputs of **Vanilla CoT** and **Fine-Tuned Visual CoT**, for problem 40. The token heat-map

is designed so that tokens with a darker background have a higher *image-to-text* attention ratio, while the tokens with a lighter background have a lower *image-to-text* attention ratio. Darker patches of ths token heatmap are areas that *attend* more to the image, compared to the problem text. It is important to note that this is a *ratio*, which means that dark patches do not correlate to simply deep, highly concentrated visual connections, but instead, they correlate to **exclusive** visual connections to image patches, *relative* to the problem text tokens (i.e. what information is only accessible in the image?). After qualitatively analyzing dozens of these model outputs, we find various validating examples of dark patches that do correlate to vision-only information, increasing our observable belief that the *image-to-text* attention ratio does contain meaningful information that may help us better understand model *reasoning*.

Overall, at a general glance, the **Fine-Tuned Visual CoT** has a more consistently darker colour scheme as the model generates its output, as agreed upon by the overall average ratios in Table 1. Beyond just a number, this is what it looks like: more consistently darker patches, which may highlight unique aspects about a reasoning sample.

We find that darker patches often correlate to 4 different types of information, most of which are exclusive to the visual component:

Geometric Properties. Referring to Figure 4, we observe that the darkest token in the output of **Fine-Tune Visual CoT** is the phrase "are parallel", which is a very crucial property to realize when solving this problem, and is only realizable by looking at the visual diagram. This word does not exist in the output of **Vanilla CoT**. We also observe that the the medium-dark patches in the fine-tuned output tend to be more explicitly, visual connected tokens such as "...because they are on the same side of the transversal" or "corresponding angles", whereas such visual detail is not as apparent in the vanilla CoT output, leading to a lighter token heat-map.

Lines. Referring to Figure 5, we can see that the darker patches in this model output appear at references to lines such as "OX" or "CD". While the problem text gives explicit details on the distances, and introduces these lines, it is hard to visualize what these lines actually look like without directly attending to the visual diagram, hence the emphasis in these locations. We also visualize the average attention maps for "O", "OX", and "CD", and see strong attention weights on the letters, as well as medium-strength patterns across the lines.

Visual Equations. Referring to Figure 6, this is an example of a problem whereby the equations themselves are only located on the image, and do not exist in the problem text. Good reasoning should effectively attend to these visual equations, transcribe them, and use them in their reasoning steps. It is clear that the **Vanilla CoT** output failed to

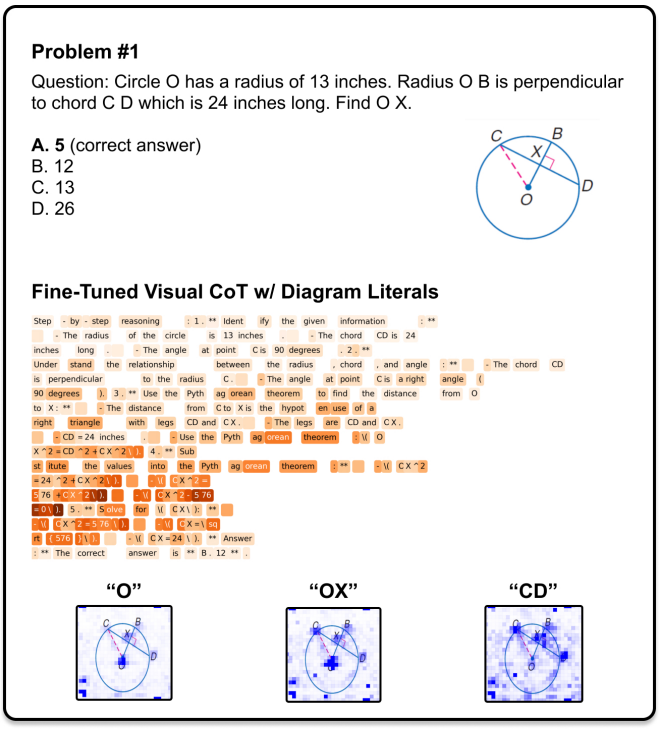

Figure 5. Token heat-map of the *image-to-text* attention ratio for problem 1. We include 3 attention maps for "O", "OX" and "CD" tokens, and overlap the attention weights onto the scaled image.

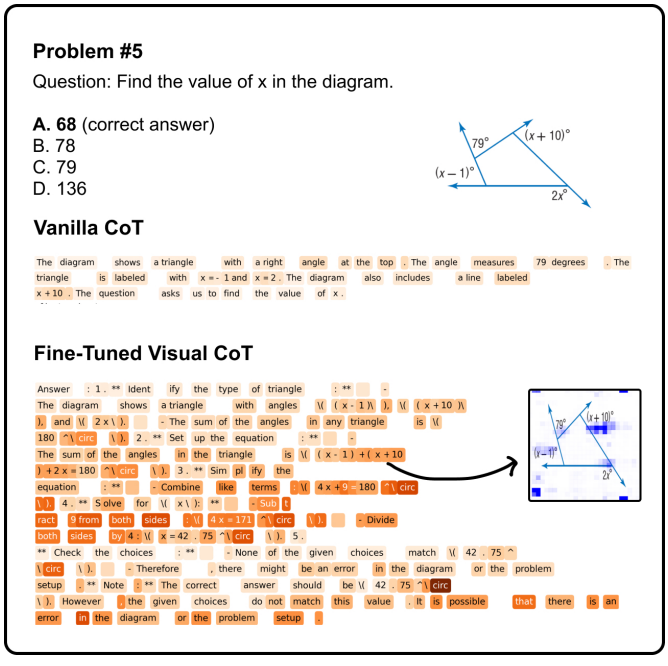

Figure 6. Token heat-map of the *image-to-text* attention ratio for problem 5. We include the visual attention map for token "+".

do so effectively, seemingly over-attending to the 'x' token and inaccurately inferring that $x = -1$ and $x = 2$, whereas

the **Fine-Tuned Visual CoT** output has a slew of medium-dark patches. The darker patches, and this is something that has been observed in many samples, tend to be the math questions themselves, which makes sense, as it is likely trying to identify what the equation is. We build the attention map for $+$ in $x + 10$, and see that it clearly tries to attend close to the patches around $x + 10$ in the image.

Reminders. Referring to Figure 8, this case is a bit different. We see that at the start of the **Fine-Tuned Visual CoT** output, there are already references to the math equations on the diagram, with a medium-dark color showing that it attended relatively exclusively to the image. However, the darkest patches are a little later, where interestingly, while the model is expected to perhaps attend to its earlier outputs (which it may very well still do), it also still attends to the image almost as a way to *remind* itself of what it saw, verifying that is correct to maximize accuracy. While there is no conclusive evidence on this behavior, we do observe that problems with vision-only math problems tend to have frequent dark patches around math equations in their output. This could be one explanation.

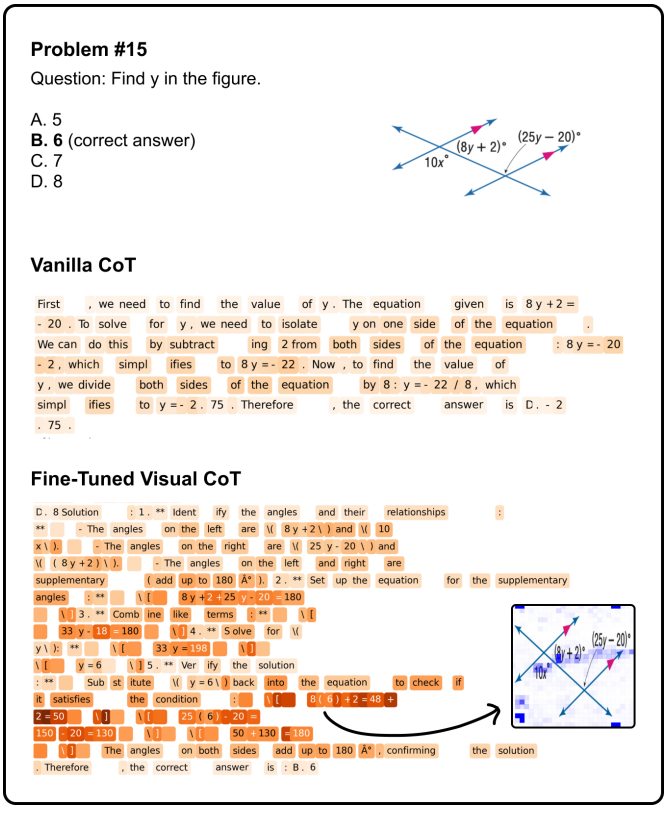

Figure 7. Token heat-map of the *image-to-text* attention ratio for problem 15. We include the visual attention map for token "6".

6.1. Limitations

One of the main limitations is that the image-to-text attention ratio is universally quite low at the start of the output sequence. This is most likely because the first few output tokens attend heavily to the input text and visual patches because they don't have much else to attend to. Over time, self-attention kicks in and the decoder has more information to make its predictions based on, but this could confound the ratios, particularly the first few output tokens. One way to overcome this is to also measure self attention at each decoder step; that is, we don't just calculate the image-to-text ratio for the input problem text, but we also calculate the extent to which each output token attends to previously generated output tokens.

Moreover, we also assume that every layer and every attention head contributes the same amount, via an averaged attention map. While we chose this approach for simplicity, there are more advanced approaches [4] [2] that more accurately aggregate across layers and heads that could build a relevancy map more directly related to an output token's connection with input tokens.

7. Conclusion & Future Work

Frontier models struggle with visual math reasoning, and yet, few efforts have gone into understanding *why*. In this study, we propose a novel approach in understanding how VLMs attend to the visual component of math problems, proposing an *image-to-text* attention ratio that shows promising potential in elucidating sections of a model's reasoning that are more vision-heavy. We demonstrate this in both quantitative and qualitative ways, both of which have implications in building greater intuition for VLMs, while creating new techniques to optimize our synthetic data generation process. We thank Zhang's work [21] for the core motivation for our hypothesis, and see our study as a valuable exploration of an approach that may tell us a little more about the way VLMs reason about visual math problems.

We have identified three main areas of future work. Firstly, while this study employed mostly qualitative approaches in evaluating the correlative ability of the *image-to-text* attention ratio, using datasets such as MathVerse [21] that explicitly label vision-only properties may allow for more rigorous correlations to be discovered. Secondly, rather than calculating this ratio for the entire model's output, breaking down a model's reasoning and measuring each step's visual emphasis could inform us of more/less impactful steps throughout a model's reasoning. This could provide a more accurate understanding of what specific type of reasoning to elicit for optimal outcomes. Finally, in the future, we hope to package our experiments and visualizations into an open-source interpret-ability tool to provide greater, qualitative insight into a model's reasoning capability.

8. Contributions & Acknowledgements

Joseph Tey, the sole author in this paper, was responsible for driving this project from start to finish. We thank the TinyLLaVA team [11] for providing the initial codebase (https://github.com/TinyLLaVA/TinyLLaVA_Factory) in which we built on top of. We used a modified version of their scripts to fine-tune our models, and we also built our interpret-ability layer over the implementation of TinyLLaVA3.1b.

References

[1] M. Abdin, S. A. Jacobs, A. A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 1

[2] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020. 2, 8

[3] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. 4

[4] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021. 3, 8

[5] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021. 2

[6] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. 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. 1

[7] J. He-Yueya, G. Poesia, R. E. Wang, and N. D. Goodman. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*, 2023. 3

[8] N. Ho, L. Schmid, and S.-Y. Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022. 2, 5

[9] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023. 1, 3

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[11] J. Jia, Y. Hu, X. Weng, Y. Shi, M. Li, X. Zhang, B. Zhou, Z. Liu, J. Luo, L. Huang, et al. Tinyllava factory: A modularized codebase for small-scale large multimodal models. *arXiv preprint arXiv:2405.11788*, 2024. 4, 9

[12] M. Jia, Z. Zhang, W. Yu, F. Jiao, and M. Jiang. Describe-then-reason: Improving multimodal mathematical reasoning through visual comprehension training. *arXiv preprint arXiv:2404.14604*, 2024. 1, 2, 3

[13] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 4

[14] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 2

[15] P. Lu, R. Gong, S. Jiang, L. Qiu, S. Huang, X. Liang, and S.-C. Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021. 3

[16] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 2

[17] G. B. M. Stan, R. Y. Rohekar, Y. Gurwicz, M. L. Olson, A. Bhiwandiwalla, E. Aflalo, C. Wu, N. Duan, S.-Y. Tseng, and V. Lal. Lvlm-intrepret: An interpretability tool for large vision-language models. *arXiv preprint arXiv:2404.03118*, 2024. 3

[18] L. Wang, Y. Hu, J. He, X. Xu, N. Liu, H. Liu, and H. T. Shen. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19162–19170, 2024. 1, 2

[19] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2, 3

[20] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 4

[21] R. Zhang, D. Jiang, Y. Zhang, H. Lin, Z. Guo, P. Qiu, A. Zhou, P. Lu, K.-W. Chang, P. Gao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*, 2024. 1, 2, 5, 8

[22] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 1, 2

[23] G. Zheng, B. Yang, J. Tang, H.-Y. Zhou, and S. Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023. 1, 2

[24] B. Zhou, Y. Hu, X. Weng, J. Jia, J. Luo, X. Liu, J. Wu, and L. Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. 4

[25] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2

9. Appendix

9.1. Inference Prompts

Vanilla Direct Inference Prompt

Please directly answer the question and provide the correct option letter, e.g., A, B, C, D.

Question: {diagram} {problem text}
Choices: {choices}

Answer:

Vanilla CoT Inference Prompt

Please first conduct reasoning, and then answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.

Question: {diagram} {problem text}
Choices: {choices}

Answer: Let's think step by step.

Fine-Tuned Inference Prompt

Solve this problem, and return the answer at the end of your response, e.g. Answer: A, B, C or D

Question: {diagram} {problem text}
Choices: {choices}

Answer:

Figure 8. Prompts used when running inference.