HoneyBee: Data Recipes for Vision-Language Reasoners

Hritik Bansal[1,2,*], **Devandra Singh Sachan**[1], **Kai-Wei Chang**[2], **Aditya Grover**[2], **Gargi Ghosh**[1], **Wen-tau Yih**[1], **Ramakanth Pasunuru**[1]

[1]FAIR at Meta, [2]University of California Los Angeles
[*]Work done at Meta

Recent advances in vision-language models (VLMs) have made them highly effective at reasoning tasks. However, the principles underlying the construction of performant VL reasoning training datasets remain poorly understood. In this work, we introduce several data curation approaches and study their impacts on VL reasoning capabilities by carefully controlling training and evaluation setups. We analyze the effects of context (image and question pair) sources, implement targeted data interventions, and explore scaling up images, questions, and chain-of-thought (CoT) solutions. Our findings reveal that (a) context source strategies significantly affect VLM performance, (b) interventions such as auxiliary signals from image captions and the inclusion of text-only reasoning yield substantial gains, and (c) scaling all data dimensions (e.g., unique questions per image and unique CoTs per image-question pair) consistently improves reasoning capability. Motivated by these insights, we introduce HoneyBee, a large-scale, high-quality CoT reasoning dataset with 2.5M examples consisting 350K image-question pairs. VLMs trained with HoneyBee outperform state-of-the-art models across model sizes. For instance, a HoneyBee-trained VLM with 3B parameters outperforms the SOTA model and the base model by 7.8% and 24.8%, respectively, on MathVerse. Furthermore, we propose a test-time scaling strategy that reduces decoding cost by 73% without sacrificing accuracy. Overall, this work presents improved strategies for VL reasoning dataset curation research.

Date: October 15, 2025
Correspondence: hbansal@g.ucla.edu

Meta

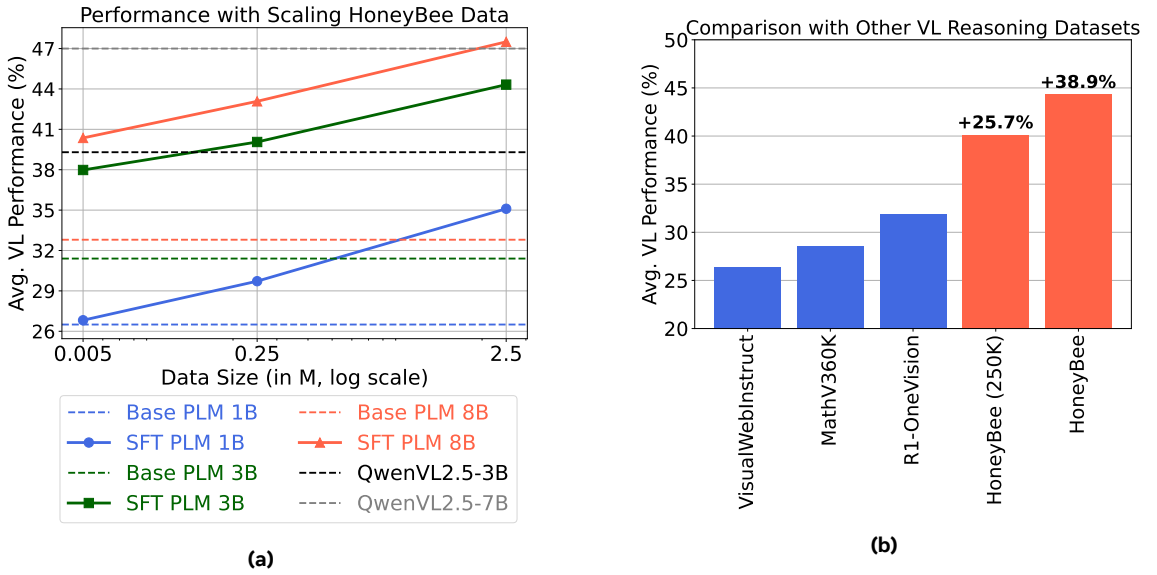


Figure 1 Summary of the results. (a) We show that training with increasing amounts of our curated HoneyBee data leads to consistent accuracy improvements averaged across five VL reasoning tasks (MathVista, MathVerse, MathVision, MMMU-Pro, and We-Math) for several model sizes (1B to 8B). (b) We report the relative gains achieved by training HoneyBee with PLM-3B compared to existing VL reasoning CoT datasets containing 250K–300K VL instances.

	Average	MathVerse (testmini, vision-only)	MathVista (testmini)	MathVision (testmini)	MMMU-Pro (vision)	We-Math (testmini)	DynaMath*	LogicVista*	Hall. Bench* (image)	MATH500*‡	GPQA*‡ (diamond)
1B model scale											
PLM-1B [13]	25.9	17.8	48.6	15.1	15.8	35.5	30.3	23.0	50.3	15.2	7.1
InternVL-2.5-1B [12]	27.6	18.7	44.2	16.4	16.2	38.7	29.3	20.5	51.6	27.8	12.1
InternVL-3-1B-Instruct [81]	28.3	18.0	35.0	13.2	16.2	37.5	28.5	21.9	56.0	37.8	19.2
PLM-HoneyBee-1B	**36.2**	**29.4**	**53.7**	**23.0**	**18.8**	**50.6**	**39.3**	**28.6**	**56.1**	**36.8**	**25.8**
3B-4B model scale											
PLM-3B [13]	33.8	18.0	57.2	16.1	19.5	46.1	37.0	33.5	61.1	30.4	18.7
InternVL-2.5-4B [12]	41.5	28.7	**61.8**	24.7	**31.1**	55.6	40.7	32.1	65.5	49.4	25.3
Qwen2.5-VL-3B-Instruct [2]	42.6	35.0	58.9	23.7	29.8	49.2	42.5	**36.6**	**66.0**	62.0	22.7
PLM-HoneyBee-3B	**46.2**	**42.8**	61.2	**29.9**	28.4	**59.3**	**51.9**	**36.6**	65.0	59.4	**27.7**
7B-8B model scale											
PLM-8B [13]	34.6	19.3	59.3	17.1	20.5	47.9	36.7	30.8	64.0	34.0	16.2
InternVL-2.5-8B [12]	41.4	27.3	61.5	21.4	32.0	56.6	41.9	26.6	63.8	57.0	26.3
InternVL-3-8B-Instruct [81]	45.1	35.3	61.8	19.4	35.8	55.7	51.2	36.2	65.5	**69.6**	20.2
Qwen2.5-VL-7B-Instruct [2]	48.5	42.0	67.5	**27.6**	**37.1**	61.1	51.3	39.9	67.4	64.8	26.3
PLM-HoneyBee-8B	**49.8**	**43.0**	**68.2**	26.3	33.8	**66.1**	**53.3**	**41.3**	**68.8**	63.6	**33.3**

Table 1 Performance of VL reasoners trained with HoneyBee data. We compare the accuracy of PLMs trained with the HoneyBee data on diverse downstream evaluation datasets. We find that models trained on HoneyBee achieve best-in-class performance across model sizes. Task-specific subsets or splits are indicated in brackets '()'. Datasets that were unseen during the data curation process are marked with *, and text-only reasoning datasets are marked with ‡.

1 Introduction

Solving reasoning problems, such as those involving mathematics in visual contexts, is a crucial capability for AI models, powering many real-world applications such as visual data analysis [43, 40], education [5], and scientific discovery [62]. Recent vision-language models (VLMs), such as GPT-4o [23], o3 [48], Gemini-2.5 [14], Llama-4 [45] achieve strong VL reasoning performance by training their pretrained models on high-quality synthetic chain-of-thought (CoT) data [77]. In particular, the ability to generate CoTs (e.g., step-by-step solution) during problem-solving enable VLMs to utilize additional inference-time computation before providing the final answer [70, 28, 20]. Yet, the multimodal CoT datasets and their recipes used for training state-of-the-art VLMs are often proprietary, leaving several open questions about their design space.

Several prior works have shown that supervised finetuning (SFT) with the quality of CoT data is crucial for LLM (text-only) reasoning performance [19, 6, 47, 59], and also serves as a key foundation for subsequent RL training [36]. However, there is a major gap in our understanding of how high-quality CoT datasets are constructed for VL reasoning, where a model needs to integrate information from several modalities (visual content in images and text content in questions) to provide accurate answers. Specifically, prior work on VL reasoning does not explore the breadth of design choices and suffers from several challenges. Firstly, the impact of context (image and question pairs) from diverse data sources remains unclear. For instance, `Math-LLaVA` [54] and `LLaVA-CoT` [73] curate different data distributions from existing image QA datasets and employ their own custom CoT generation strategies. Thus, it is uncertain how much of the reasoning performance of models trained on these datasets can be attributed to the quality of the context. Secondly, prior work [33, 21, 9] suggests that targeted data interventions—such as visual perturbations and difficulty filtering—can enhance model perception and problem-solving. However, there has been limited exploration of other interventions that could further improve data quality. Thirdly, while scaling up the training data is known to enhance reasoning performance [19], it remains unclear whether VL reasoning data should be scaled along specific axes or across all axes, such as the number of images, the number of questions per image, and the number of CoTs per (image, question) pair.

Importantly, there is a lack of fair and robust comparisons between diverse design decisions. For fairness, the direct effect of a design choice should be measured by fixing the training setups (e.g., SFT starting from

identical models) and the evaluation protocol. For robustness, models should be trained at multiple scales (such as 3B and 8B) and evaluated on a collection of datasets rather than a single dataset. To address these critical questions in VL reasoning data design, we adopt a comprehensive and scalable approach to identify the key factors in dataset curation (Figure 2). This effort culminates in the creation of **HoneyBee**, a high-quality and one of the largest VL reasoning CoT datasets, comprising **2.5M** (image, question, CoT) tuples.

In our data curation process, we first study the impact of diverse contexts (i.e., image and question pairs) acquired from several VL reasoning datasets and rank them based on the performance of multiple VLMs (i.e., 3B and 8B models) on a battery of evaluation datasets. Interestingly, we observe that the choice of source datasets can lead to significant differences in model performance, with up to a 4% difference in average accuracy across evaluation datasets (§3.1). In the next stage, we create a list of data enhancement strategies, including *visual perturbation*, *text-rich images*, *perceptual redundancy*, *shallow perception*, *caption-and-solve*, *text-only reasoning*, *increased distractors*, *length* and *difficulty* filtering, applied on top of the best-performing dataset from the previous stage (Figure 3). Our experiments show that several data interventions fail to outperform the baseline dataset, highlighting their limited practical value despite strong motivations. Importantly, we reveal that auxiliary signals from image captions (*caption-and-solve*) and augmenting the VL reasoning data with text-only reasoning data lead to major improvements across diverse benchmarks (§5.2). In our experiments, we find that model performance improves with scaling all dimensions (images, questions, and CoTs) in the reasoning data (§5.3).

Motivated by these findings, we construct the HoneyBee dataset, and train VLMs of several sizes (1B–8B) on the HoneyBee dataset. We show that reasoning performance strongly scales with the amount of training data and outperforms existing state-of-the-art instruction-tuned VLMs, indicating at the quality and scalability of our dataset (§5.4). In particular, PLM-HoneyBee-1B achieves a relative performance improvement of 28 percentage points (pp) over InternVL-3-1B-Instruct, averaged across ten evaluation datasets (Table 1). Furthermore, PLM-HoneyBee-3B and PLM-HoneyBee-8B achieve relative gains of 8.4pp and 2.7pp over Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-8B-Instruct, respectively. Ultimately, we also propose an efficient decoding strategy for test-time scaling that enables the generation of multiple solutions from the HoneyBee-trained VL reasoners, using 73% fewer inference tokens without any loss in performance (§ 5.5). Our thorough experiments yield several insightful findings that lay the foundation for curating the next generation of VL reasoning datasets.

2 Preliminaries

In this work, we focus on the curation of high-quality, large-scale synthetic data to enable strong vision-language reasoning capabilities. Let $\mathcal{D} = \{(I_j, Q_j, A_j)\}_{j=1}^{N}$ denote the source vision-language reasoning dataset of size N, where each entry consists of an image I_j, a corresponding question Q_j, and an optional final answer A_j.[1] Further, let $\mathcal{G}$ be a synthetic data generator that outputs a textual chain-of-thought (CoT) to solve the questions about the images in $\mathcal{D}$, such that $C_j = \mathcal{G}(I_j, Q_j)$. In particular, the CoT C_j is composed of several reasoning steps, including step-by-step solutions, planning, self-verification, and self-reflection behaviors [58], denoted as S_j. This is followed by a predicted final answer P_j to the given question about the image. Thus, $C_j = [S_j; P_j]$, where ; denotes concatenation in the raw text space. Thus, We represent the synthetic data as $\mathcal{D}_{\mathcal{G}} = \{(I_j, Q_j, C_j, A_j)\}_{j=1}^{N}$.

Given synthetic data, we train a VLM (p_θ), using a supervised finetuning objective: $\mathbb{E}_{(I_j,Q_j,C_j)\in\mathcal{D}_{\mathcal{G}}}[\log p_\theta(C_j \mid I_j, Q_j)]$, i.e., maximizing the probability of the problem-solving CoT given the image and question as context. Post-training, the VLM will perform step-by-step reasoning before generating its predicted answer, thus utilizing additional test-time compute [70]. Overall, the goal of high-quality synthetic data is to train performant VL reasoners that can solve novel problems from diverse downstream tasks such as geometry, function plots, and charts [40]. In this work, we focus on the paradigm of training on smaller VLMs on synthetic data from larger generator models. This approach is more compute-efficient, more popular [54, 19, 6, 32], and allows for comprehensive training runs on diverse synthetic data distributions. In practice, p_θ can be a weaker VLM reasoner than the generator [54], a setup commonly referred to as knowledge distillation [22], where a strong teacher guides a weak student. Alternatively, the generator itself can be used as the student, a process

[1] When there is no final answer, we can set $A_j = \phi$.

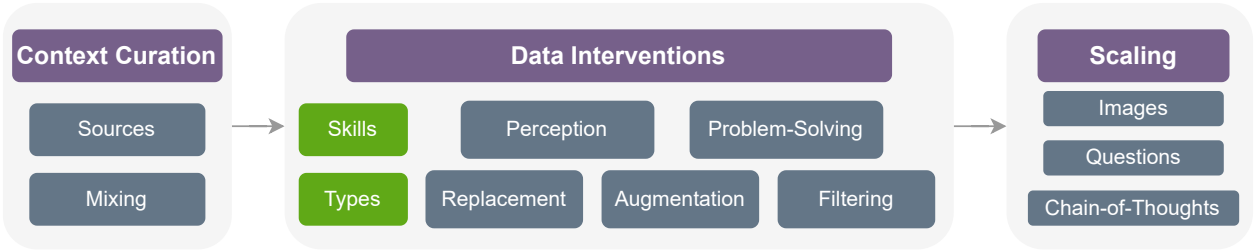


Figure 2 Overview of the data curation pipeline. In this work, we curate the context (image, question) from diverse sources and assess the impact of their mixing. Further, we curate a set of data interventions that target diverse skills and types. Subsequently, we study the impact of scaling along different data axes. These insights lead to the creation of our large-scale HoneyBee dataset.

known as self-improvement [55]. Prior research has also explored training a stronger reasoner with data from a weaker one, referred to as weak-to-strong reasoning [8, 75, 4].

3 Data Curation Pipeline

We outline our vision-language reasoning data curation pipeline (Figure 2), which consists of multiple stages: (a) `context curation`, which assesses the impact of diverse context (image, question) data sources, as well as their mixing (§3.1); (b) `data interventions`, which aim to enhance perception and problem-solving skills to enable strong VL reasoning capabilities (§3.2); and (c) `scaling`, which studies the impact of scaling diverse components of the reasoning data (§3.3).

3.1 Context Curation

The quality of context i.e., image, question pair, is crucial for determining the knowledge, and reasoning skills imparted to the VL reasoner. For instance, a VL reasoner exposed to diverse geometric images and questions will excel in downstream geometry problem-solving [80]. Thus, we aim to curate contexts that allow the resulting VL reasoner to achieve good performance on several downstream reasoning tasks.

Sourcing. In the past few years, several CoT datasets have been proposed to enable strong vision-language (VL) reasoning capabilities [54, 24, 73]. However, the reported improvements are a combination of several factors, such as the contexts present in the dataset, custom training algorithms, and the quality of the generator model. Furthermore, the contexts in different datasets originate in diverse ways. In particular, the contexts in reasoning datasets like `Math-LLaVA` and `R1-OneVision` [54, 74] are composed of several existing expert-curated and task-specific datasets, such as Geometry-3K [38], IconQA [39], CLEVR-Math [35], while `ViRL` [65] integrates several VL datasets [16] and performs a customized cleaning process. Similar to prior data curation work [19, 31], we fix the training algorithm (e.g., supervised finetuning) and the CoT generator model to analyze the *direct effect* of the contexts in individual datasets on training performant VL reasoners.

First, we list several widely adopted VL reasoning datasets consisting of (image, question, final answer) tuples. Next, we remove instances from these datasets that contain images with identical perceptual hashes[2] to any of the images in the evaluation VL reasoning tasks [82]. Second, we prompt our generator model to produce a CoT to solve the (image, question) pairs in these datasets. Importantly, we consider small, similarly sized subsets of these datasets to control for the impact of data quality, rather than data quantity, on model performance. Similar to [19], this approach also facilitates faster training iterations. Further, we also create a filtered version of each dataset by discarding the CoTs that do not lead to the correct final answer. This step ensures that we do not penalize a dataset when our generator produces an incorrect solution. Third, we train multiple VLMs (e.g., 3B and 8B models) on different context sources (and their filtered versions) and evaluate them on several downstream VL reasoning tasks. Finally, we rank each individual source dataset based on the average performance across multiple downstream tasks and model trainings.

[2] https://github.com/idealo/imagededup

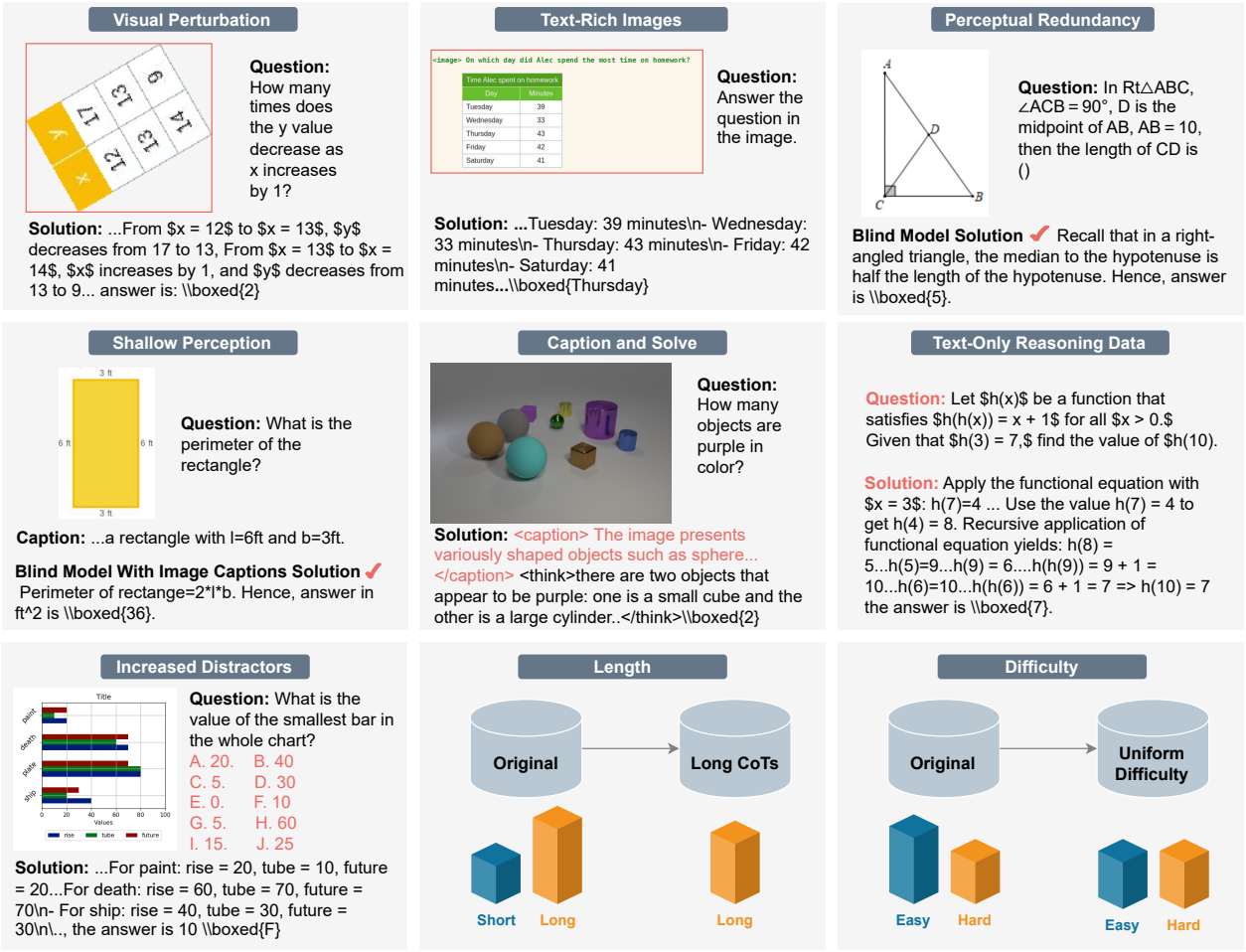


Figure 3 Overview of data intervention strategies. In this work, we curate a diverse set of data interventions to enhance the quality of VL reasoning CoT data. These methods target a range of skills, such as perception and problem-solving capabilities. We present details for each intervention in §3.2.

Mixing. Here, we consider mixing as an additional strategy to combine the strengths of the top-performing data sources. Specifically, prior work in LLM reasoning [19, 29] has shown that mixing can yield superior datasets compared to using individual sources alone. To this end, we mix the data from the top-2, top-4, and all datasets from the previous stage. Subsequently, we take an equally sized subset from each mixture to control for dataset size and focus solely on mixture quality. We then train multiple VLMs on these mixtures and evaluate them on a battery of VL reasoning tasks. Finally, we assess whether any of the mixtures achieve better average performance than the best-performing individual dataset.

3.2 Data Interventions

Starting with the best data mixture from the previous stage, we assess whether its quality can be further enhanced through targeted data interventions (Figure 3). Specifically, these interventions aim to improve particular skills of the VL reasoner, including *perception* and *problem-solving* capabilities. Enhanced perception is crucial for robust visual understanding of image content within the model's context, and strong problem-solving ability is essential for accurate step-by-step solutions and calculations.

In addition, the data intervention strategies can affect the original VL CoT reasoning data in various ways, including *replacement*, *augmentation*, and *filtering*. The replacement strategy substitutes part (or all) of the original dataset with higher-quality variants, without changing the overall size of the dataset [42]. In contrast, the augmentation strategy increases the amount of training data by either adding transformed variants of the

original data [78] or introducing external knowledge [3]. Finally, we also consider diverse filtering scenarios, which improve the quality of the original data by removing lower-quality instances according to predefined rules or classifiers [17].

3.2.1 Perception Enhancement

First, we will outline the various data interventions for enhancing the *perception* of the VL reasoner:

Visual Perturbation. This intervention creates a perturbed version of the original image in the reasoning dataset to enhance the perceptual robustness of the VLM [33]. Starting from an instance in the synthetic reasoning dataset, $\{I_j, Q_j, C_j\} \in \mathcal{D}_{\mathcal{G}}$, we apply one of three transformations, $\mathcal{T} = \{$*rotation*, *distractor concatenation*, *dominance-preserving mixup*$\}$, randomly to the image in each instance such that $I'_j = h(I_j)$, where h is a transformation from $\mathcal{T}$. We test the impact of this method in two ways: (a) as a *replacement*, where we perturb the images in part of the dataset while leaving the other part untouched, and (b) as an *augmentation*, where we mix the transformed dataset with the original dataset.

Text-Rich Images. It is crucial for VLMs to comprehend textual information embedded in images to perform accurate reasoning [76, 64]. To enhance the model's ability to read and integrate textual information within images, we programmatically transform each (image, question) pair from the original data into a new image. Specifically, the new dataset $\mathcal{D}_{TR}$ contains only the transformed text-rich image and the CoT from the original dataset. In particular, we embed the question in diverse font styles and colors, overlaying it with the original image on a blank background with various colors. Formally, $I'_j = Render(I_j, Q_j)$, thus, the new dataset is $\mathcal{D}_{TR} = \{I'_j, C_j\}_{j=1}^{N}$. Similar to visual perturbation, we assess the usefulness of this method as both a *replacement* and an *augmentation* strategy.

Perceptual Redundancy. Prior work [79] has shown that VLMs often rely on the visual content embedded within textual questions for answer deduction, rather than truly understanding the context image. Additionally, we observe that some textual questions can be answered accurately without access to the image, using only the model's existing knowledge. For instance, a strong VLM can utilize its pretrained knowledge to answer questions (e.g., `What will lead to an increase in the population of deer? i. increase in lion, ii. decrease in plants, iii. decrease in lion, iv. increase in pika`) without access to the food web image [27]. To encourage greater reliance on visual inputs, we *filter* the original synthetic dataset by removing instances where the generator model leads to the correct final answer without access to the image; that is, we exclude examples where the predicted answer $\hat{P}_j = \mathcal{G}(Q_j)$ matches the ground-truth answer A (i.e., $\hat{P}_j = A_j$).[3]

Shallow Perception. In the existing VL datasets, there are several instances that can be solved by querying the generator to solve them with access to the question and an image caption. Since image caption contains shallower information than all the visual content in the original image, such instances will not require deep visual insights in the reasoning process. Thus, we filter the original dataset where the generator model leads to the correct final answer with access to the question and image caption but not the image; that is, we exclude examples where the predicted answer $\hat{P}_j = \mathcal{G}(Q_j, I_j^{cap})$ matches the ground-truth answer A_j (i.e., $\hat{P}_j = A_j$). In our experiments, the image caption $I_j^{cap} = \mathcal{G}(I_j)$ is also generated by the same generator model.

Caption and Solve. We explore an active approach to enhancing the visual understanding capabilities of the VL reasoner by providing auxiliary visual signals from the image caption I_j^{cap} from the stronger generator model. Specifically, we augment the original dataset's CoT by including the image caption in the training data, i.e., $C'_j = [I_j^{cap}; C_j]$, where ; denotes concatenation in the raw text space. The transformed reasoning dataset is then $\mathcal{D}_{CS} = \{I_j, Q_j, C'_j\}_{j=1}^{N}$. By design, this method also increases the length of the CoT, $|C'_j| > |C_j|$, which facilitates the use of inference-time compute. We provide more information about different configurations in Appendix B.

[3] In our experimental setup, we use the same foundation model as CoT generator as well as caption generator. Thus, we take the liberty to denote image captioner as $\mathcal{G}$.

3.2.2 Problem-Solving Skill Enhancement

Next, we will outline the various data interventions for enhancing the *problem-solving* skills of the VL reasoner:

Text-Only Reasoning. Through this intervention, we aim to enhance problem-solving skills by exposing the VLM to novel problems and solutions from a high-performing text-only reasoning dataset [19]. This strategy serves multiple purposes: (a) it allows the VLM to learn useful skills from existing unimodal reasoning data via cross-task transfer, and (b) it makes the VLM a more general-purpose reasoner, enabling it to solve textual reasoning problems too beyond VL reasoning problems. In practice, this method acts as an *augmentation*, where the final dataset is a concatenation of the original VL reasoning data and the high-performing text-only reasoning data, i.e., $\mathcal{D}_{\mathcal{G}}^{VL+Text} = \mathcal{D}_{\mathcal{G}}^{VL} \cup \mathcal{D}_{\mathcal{G}}^{Text}$. For consistency, we re-annotate the problems from the text-only reasoning data to obtain unimodal CoTs from the same generator model.

Increased Distractors. Prior work [69, 26, 76] has shown that introducing distractors in reasoning tasks, specifically by increasing the number of options (e.g., from four to ten), makes them harder for reasoning models to solve. Motivated by this observation, we explore a strategy to transform the original questions to have ten options and modify the original CoT accordingly; that is, $(Q'_j, C'_j) = \text{LLM}(Q_j, C_j)$. This method aims to encourage the VLM not to rely on chance when answering complex problems and to boost model robustness. We test the impact of this method in two ways: (a) as a *replacement*, where part of the original dataset is transformed with distractors while the other part remains untouched, and (b) as an *augmentation*, where we mix the transformed dataset with the original dataset.

Length. In this method, we aim to encourage a reasoner to generate longer chains of thought (CoTs). This approach allows the model to utilize additional test-time compute for more detailed visual comprehension, stepwise solutions, planning, and re-evaluation during problem-solving. To achieve this, we compute the distribution of CoT lengths in the original reasoning dataset and split it into two equal halves: those with less than and those with more than the median CoT length. Subsequently, we train the models on the longer half to promote extended reasoning behavior in downstream tasks.

Difficulty. Reasoning datasets are typically skewed toward easier problems, as these are easier to acquire at scale compared to more difficult problems, which only a limited number of experts can solve [21]. To increase the difficulty of our dataset, we first classify all instances into diverse difficulty levels (1: beginner, 2–3: AMC, 4: intermediate level AIME, >4: Olympiad level) using an LLM and a predefined rubric.[4] We then filter the dataset to ensure that the representation for each level is roughly balanced. This enables the model to learn from harder examples and tackle more complex VL reasoning problems in downstream tasks.

We present additional details about the data intervention strategies in Appendix B. In theory, all the data intervention strategies listed above should enhance VL reasoning behaviors. However, an intervention strategy may steer the model toward behaviors that benefit a specific evaluation dataset at a particular model scale. Thus, outperforming a simple baseline of training on the original reasoning dataset across various downstream VL reasoning tasks and multiple model scales remains a non-trivial challenge. Therefore, it is essential to compare these strategies under identical conditions (e.g., training and evaluation) to identify those that consistently improve performance across the board.

3.3 Scaling Diverse Data Axes

Scaling the amount of reasoning CoT data has consistently helped in enhancing the LLM performance on several downstream tasks [19, 60]. However, there has been limited exploration on the impact of scaling data for VL reasoning. In addition, the introduction of a new modality (image) in VL space adds a new dimension to scaling training data. Hence, we study the impact of scaling diverse axes (e.g., image, question, and CoT) on the reasoning capabilities of the VLM. Here, we explain the method to study the scaling behaviors starting from a dataset containing N unique images, and corresponding question and CoT $\mathcal{D}_N = \{I_j, Q_j, C_j\}_{j=1}^{N}$. We drop the subscript $\mathcal{G}$ for simplicity.

[4] https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings

Scaling Images. We create three additional subsets of the original dataset, each of increasing size, resulting in four datasets in total: $[\mathcal{D}_{N/8}, \mathcal{D}_{N/4}, \mathcal{D}_{N/2}, \mathcal{D}_N]$. Here, $\mathcal{D}_{N/8}$ indicates that we randomly select $N/8$ instances from the original dataset. As we increase the data size, we are effectively scaling the number of unique images in the dataset. Subsequently, we train several VLMs on these datasets and study whether the average performance on downstream tasks improves with increased data scale.

Scaling Questions. We study the impact of synthesizing novel questions by using the existing questions as seeds. In particular, we prompt the question generator to create new questions that are reasonable, solvable, and of similar difficulty to the original questions, conditioned on the image and question [60]; i.e., we generate $Q'_{j,i} = \mathcal{G}(I_j, Q_j)$, where $Q'_{j,i}$ is the i^{th} synthetic question for (I_j, Q_j) using the question generator $\mathcal{G}$.[5] Subsequently, we create a CoT for the newly synthesized questions using the generator model. We then merge this synthetic data with the original data, thereby scaling the dataset by having multiple questions for a given image. In our experiments, we construct four subsets of synthetic reasoning data: $[\mathcal{D}_{N/8}(n_q = 1), \mathcal{D}_{N/8}(n_q = 2), \mathcal{D}_{N/8}(n_q = 4), \mathcal{D}_{N/8}(n_q = 8)]$, where $\mathcal{D}_N(n_q = k) = \{I_j, Q_j, C_j\}_{j=1}^{N} \cup \{I_j, \{Q'_{j,i}, C_{j,i}\}_{i=1}^{i=k-1}\}_{j=1}^{j=N}$ and $C_{j,i} = \mathcal{G}(I_j, Q'_{j,i})$. If $n_q = 4$, then three questions are newly synthesized, and the remaining question comes from the original data. We present the prompt used to generate new questions and provide qualitative examples in Appendix E and Appendix F, respectively.

Scaling CoTs. We study the impact of increasing the number of CoT traces for a given image and question pair in the reasoning dataset. This approach exposes the VL reasoner to diverse problem-solving strategies for a given problem. Specifically, we create four subsets of the dataset with an increasing number of CoTs per (image, question) pair: $[\mathcal{D}_{N/8}(n_c = 1), \mathcal{D}_{N/8}(n_c = 2), \mathcal{D}_{N/8}(n_c = 4), \mathcal{D}_{N/8}(n_c = 8)]$, where $\mathcal{D}_N(n_c = k) = \{I_j, Q_j, \{C_{j,i}\}_{i=1}^{k}\}_{j=1}^{N}$. Ultimately, we combine the findings from context curation, data interventions, and scaling diverse modalities experiments to create one of the largest VL reasoning datasets, HoneyBee (see §5.3 for more details).

4 Experimental Setup

Training Data. Our goal is to collect a diverse datasets as sources of (image, question, final answer) tuples. This data will be used to create the multimodal CoT with our generator model. Specifically, we use six source datasets: `ViRL` [65], `Math-LLaVA` [54], `R1-OneVision` [74], `ThinkLite-VL-Hard` [67], `LLaVA-CoT` [73], and `MMK12` [44]. We then perform decontamination to remove any identical images from the evaluation datasets using exact deduplication with the pHash algorithm. We cap the number of instances at 50K to focus on the impact of data quality in our data curation experiments. Subsequently, we train VL reasoners for 5 epochs across all experiments, and select the best-performing checkpoint (out of 5) based on the average performance across five downstream tasks. More details about the data statistics are provided in Appendix Table 7.

Generator Model. We fix the CoT generator model to a performant VLM, `Llama4-Scout` [45]. Specifically, it is an open-weights model consisting of 109B total parameters, of which 17B are active. This model enables efficient inference on a single A100 node using vLLM [63]. We use a non-API model to avoid changes in the quality over time [10], and powerful VLM that the community can host locally with accessible compute.

Models. We train the 3B and 8B Perception Language Models (PLMs) [13] for all data curation experiments. PLMs cannot generate vision-language CoTs, due to lack of appropriate instruction data. Thus, they serve as good base models that can be converted into VL reasoners with high-quality reasoning CoT data. Once our final data is ready, we also train much smaller VL reasoners (i.e., PLM-1B). We present more details about the training setup in Appendix D.

Evaluation. For our data curation experiments, we choose *five* VL reasoning downstream tasks as validation datasets for hill-climbing. These include `MathVerse` (testmini, vision-only subset) [79], containing 788 examples on geometry and functions; `MathVista` (testmini) [40], containing 1000 examples on diverse visual scenarios

[5] In our experiments, we use the same foundation model for both CoT generation and question generation; hence, we denote the question generator as $\mathcal{G}$.

Context Sources	Models	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
ViRL † [65]	3B	38.8	32.9	58.6	25.7	25.0	52.0
	8B	41.3	34.5	64.5	23.4	28.2	55.8
	Avg.	**40.1**	**33.7**	**61.6**	24.5	**26.6**	**53.9**
Math-LLaVA [54]	3B	36.3	30.0	56.3	23.6	21.7	50.1
	8B	39.2	33.7	62.2	23.4	26.1	50.5
	Avg.	37.7	31.8	59.2	23.5	23.9	50.3
R1-OneVision [74]	3B	35.7	30.6	55.2	20.1	22.8	49.7
	8B	38.8	32.5	60.5	22.0	26.9	52.2
	Avg.	37.3	31.6	57.9	21.1	24.9	51.0
ThinkLite-VL-Hard † [67]	3B	34.6	25.5	53.7	25.3	20.6	48.1
	8B	39.5	31.2	61.2	24.0	27.7	53.1
	Avg.	37.1	28.4	57.5	**24.7**	24.2	50.6
LLaVA-CoT [73]	3B	34.6	28.5	54.2	19.0	22.6	48.6
	8B	37.9	32.7	60.3	18.7	26.5	51.5
	Avg.	36.3	30.6	57.3	18.9	24.5	50.1
MMK12 [44]	3B	34.6	28.6	58.1	19.0	26.4	40.6
	8B	37.3	29.2	55.1	24.3	26.4	51.5
	Avg.	36.0	28.9	56.6	21.7	26.4	46.1

Table 2 Ranking the quality of the context from diverse datasets. We train PLM-3B and PLM-8B on CoTs generated on the context (image, question) pairs from diverse dataset. Then, we rank them based on the average performance of these models on several VL reasoning downstream tasks. We mark the datasets which benefit from final answer correctness filtering in these experiments by †.

(e.g., tables, functions, geometry, papers, and IQ tests); MathVision (testmini) [66], containing 304 examples sourced from math competitions of various difficulties; MMMU-Pro (vision) [76], containing 1730 examples from diverse subject areas (e.g., art and design, science, humanities, and engineering); and We-Math (testmini) [51], consisting of 1740 examples focused on diverse knowledge granularity. Overall, we track the accuracy score averaged over these five evaluation datasets and two model trainings (PLM-3B and PLM-8B) during the data curation process.

After the creation of our final HoneyBee dataset, we also evaluate them on *five* more unseen evaluation datasets including DynaMath [83] and LogicVista [71] VL reasoning datasets containing 1000 and 448 examples, respectively. Further, we evaluate the visual understanding capabilities on HallusionBench [18], a challenging perception dataset. In addition, we evaluate the general-purpose reasoning capabilities of a VLM on text-only reasoning datasets including math-centric MATH500 [34] and science-centric GPQA [52] containing 500 and 198 examples, respectively. To ensure consistency, we use *accuracy* as the scoring metric and an identical prompt, instructing the model to always answer in `boxed` format, across all evaluations. Further, we use greedy decoding with maximum generation length of 2048 across all the evaluation datasets.

5 Experiments

5.1 Impact of Context Curation

Sourcing. We present results for training PLM-3B and PLM-8B on reasoning data constructed using diverse context sources in Table 2. Specifically, we compute the average performance across five VL reasoning datasets and rank the context sources from highest to lowest accuracy. For each context source, we report the best results achieved from one of two scenarios: unfiltered CoT data and filtered CoT data based on final answer correctness.[6] Our experiments reveal that the choice of context source has a significant impact on downstream VL reasoning performance. Specifically, we observe a relative gap of 11.4 percentage points (pp) between the average performance of the lowest (MMK12, 36.0%) and highest performing source datasets (ViRL, 40.1%). This highlights the choice of context source has a significant impact on downstream VL reasoning performance.

[6]We find that ViRL and ThinkLite-VL-Hard benefit from the final answer correctness filter, while the others perform better with the unfiltered dataset (Appendix Table 8).

Mixing Sources	Models	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
Top 1	3B	38.8	32.9	58.6	25.7	25.0	52.0
	8B	41.3	34.5	64.5	23.4	28.2	55.8
	Avg.	**40.1**	33.7	61.6	24.5	26.6	53.9
Top 2	3B	37.8	33.4	58.9	21.1	25.6	50.2
	8B	39.3	32.7	60.9	20.7	29.2	53.0
	Avg.	38.6	33.1	59.9	20.9	27.4	51.6
Top 4	3B	36.7	31.5	57.6	20.1	24.0	50.5
	8B	39.7	34.9	61.6	23.4	28.2	50.3
	Avg.	38.2	33.2	59.6	21.7	26.1	50.4
All	3B	36.5	33.1	58.3	17.8	23.2	49.9
	8B	40.5	35.8	61.7	25.3	28.2	51.3
	Avg.	38.5	34.5	60.0	21.5	25.7	50.6

Table 3 Impact of mixing source datasets. We mix the VL reasoning CoT data from diverse datasets to assess whether it leads to better performance than the individual datasets itself. We find that the individual dataset source, `ViRL`, achieves the best performance.

Mixing. We next assess the impact of mixing contexts from different data sources. To this end, we combine examples from the previous stage for the top-2, top-4, and all source datasets, and select a random subset of size 50K. The results of training PLM-3B and PLM-8B on these mixtures are presented in Table 3. Interestingly, we find that mixing data from diverse sources does not improve the performance over the best-performing dataset, `ViRL`. This suggests that mixing instances from diverse sources can degrade the performance on the VL reasoning tasks.[7]

5.2 Impact of Data Interventions

Starting from the best data from the previous stage, we note the impact of several data intervention experiments on VL reasoning capabilities in Table 4. Specifically, we target the strategies at improving the perception and problem-solving capabilities of the VLMs.[8]

We find that the best-performing data intervention for enhanced perception is the *caption and solve* strategy, which provides auxiliary visual signals about the image description in the reasoning CoT data (i.e., `<caption> description </caption> solution` $\boxed{answer}$). Specifically, this approach improves accuracy averaged across multiple downstream tasks and model training runs by 3.3pp. In addition to performance gains, we show that the ability to generate image description before problem-solving offers novel efficiency improvements in test-time scaling scenarios, such as generating multiple solutions per (image, question) pair (§5.5).

Data	Avg.	PLM-3B	PLM-8B
Baseline	32.2	30.4	34.0
VL	39.2	34.6	43.8
Text	62.7	60.2	65.2
VL + Text	59.7	57.0	62.4

Table 5 Generalization to MATH500. We present the performance of VL reasoners on text-only MATH500 reasoning task. The baseline refers to the base models performance, VL refers to exposure to just VL reasoning data, Text refers to training with re-annotated `OpenThoughts3`, and VL+Text refers to mixture of VL and text reasoning CoT data.

In addition, we observe that the best-performing data intervention for enhanced problem-solving is the inclusion of *text-only reasoning* data in the training mixture. In particular, we observe a relative accuracy improvement of 7.5pp over the original dataset across multiple VL reasoning tasks and training runs. Apart from improving the VL reasoning, we want our VL reasoners to be general-purpose reasoners, thus, we study the performance of VLMs on blind (text-only) reasoning evaluation dataset `MATH500` in Table 5.

Specifically, we find that the base models PLM-3B and PLM-8B do not perform well on the `MATH500` y

[7]We leave the exploration of complex data mixing strategies [72] for VL reasoning as future work.

[8]We present the results for the best-performing configuration for the interventions. For instance, we experiment with diverse ways of implementing *caption and solve* method, and show the results for the best-performing variant.

Method	Skill	Type	Models	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
Original Data	-	-	3B	38.8	32.9	58.6	25.7	25.0	52.0
			8B	41.8	35.3	65.8	22.4	29.8	55.6
			Avg.	40.1	33.7	61.6	24.5	26.6	53.9
Caption and Solve	Perception (Auxiliary Signal)	Augment	3B	39.7	35.7	56.6	24.7	26.7	55.0
			8B	43.0	38.3	63.2	26.3	30.4	56.9
			Avg.	41.4 (+3.3pp)	37.0	59.9	25.5	28.6	56.0
Visual Perturb	Perception (Robustness)	Replace	3B	36.5	31.2	57.5	18.1	25.8	49.9
			8B	40.5	31.9	63.4	22.7	29.5	54.8
			Avg.	38.5	31.6	60.5	20.4	27.7	52.4
Text-Rich Images	Perception (Synthetic Images)	Replace	3B	37.1	30.2	55.5	22.7	27.1	49.9
			8B	40.5	35.3	61.0	21.1	30.6	54.5
			Avg.	38.8	32.8	58.3	21.9	28.9	52.2
Perceptual Redundancy	Perception (Feasibility wo/ image)	Filter	3B	36.0	31.2	54.6	21.4	24.6	48.4
			8B	36.9	32.7	58.0	18.1	27.8	48.1
			Avg.	36.5	32.0	56.3	19.8	26.2	48.3
Shallow Perception	Perception (Feasibility w/ caption)	Filter	3B	34.7	29.3	53.4	20.7	23.8	46.2
			8B	36.5	29.2	57.3	21.4	28.1	46.3
			Avg.	35.6	29.3	55.4	21.1	26.0	46.3
Text-Only Data	Problem-Solving (Cross-modal transfer)	Augment	3B	41.9	38.1	60.1	25.7	26.8	58.9
			8B	44.2	41.2	63.4	27.3	31.4	57.6
			Avg.	43.1 (+7.5pp)	39.7	61.8	26.5	29.1	58.3
Increased Distractors	Problem-Solving (Robustness)	Replace	3B	33.1	23.4	51.3	19.7	22.3	48.8
			8B	36.2	27.4	53.2	20.1	27.5	52.6
			Avg.	34.6	25.4	52.3	19.9	24.9	50.7
Length	Problem-Solving (Long Thinking)	Filter	3B	33.9	28.3	50.3	19.7	24.6	46.8
			8B	38.8	32.6	60.7	19.4	28.3	53.2
			Avg.	36.4	30.5	55.5	19.6	26.5	50.0
Uniform Difficulty	Problem-Solving (Hardness)	Filter	3B	33.1	25.1	52.9	17.4	24.1	45.8
			8B	36.1	28.6	56.8	21.4	24.9	48.8
			Avg.	34.6	26.9	54.9	19.4	24.5	47.3

Table 4 Results for data intervention experiments. We compare the performance of VL reasoners trained on datasets created using diverse data intervention strategies. Our results show that achieving better performance than the original data, which is obtained by selecting the right data sources and their mixtures, is non-trivial. We also report the results for the best-performing configuration from each intervention strategy. We find that augmenting the original CoT with image captions (*caption and solve*), and mixing with *text-only reasoning* data reliably improve VL reasoning performance.

achieving an accuracy of 32.2% and 34.0%, respectively. We find that training these models with VL reasoning CoT data (baseline data in Table 4) improves the average performance to 39.2%, indicating cross-modal reasoning transfer (vision-language to language-only). However, this performance still lags behind training the VLMs with text-only reasoning data (`OpenThoughts3`). But, training with the mixture of VL and text-only reasoning data improves the accuracy from 39.2 to 59.7 while improving the VL reasoning performance too. Thus, we will apply *caption and solve* and *text-only reasoning* data intervention strategies to create our final VL reasoning data. Interestingly, we find that most methods lead to poorer performance than the baseline, which suggests that it is non-trivial to improve over the best-chosen data source using the targeted data interventions.

5.3 Scaling to Million Samples

Scaling Trends Across Data Axes. To synthesize large-scale data, we study the scaling behavior of the best-performing dataset, `ViRL`, including the number of unique images, the number of questions per image, and the number of CoTs per (image, question) pair. We present the results for accuracy averaged across the validation downstream datasets in Figure 4. Interestingly, we find that the performance of the VL reasoners improves with scaling in images, new questions, and CoTs across model trainings of both PLM-3B and PLM-8B.

Putting Everything Together. Firstly, we include all the real data in the `ViRL` datasets, which consists 39K (image, question, and final answer) tuples. Since the amount of real data is limited, we scale the data by generating several CoTs for all (image, question) pairs, and synthetically creating new questions (Figure 6). Specifically, we generate 16 CoTs per real image and existing question pair (*scaling CoTs*). To retain the highest quality data, we filter out CoTs that do not lead to the correct final answer, leading to roughly 400K instances. To scale questions, we 14 new questions per image, resulting in 15 questions per image in total (*scaling questions*). However, we do not have access to the final answers when generating new questions. To

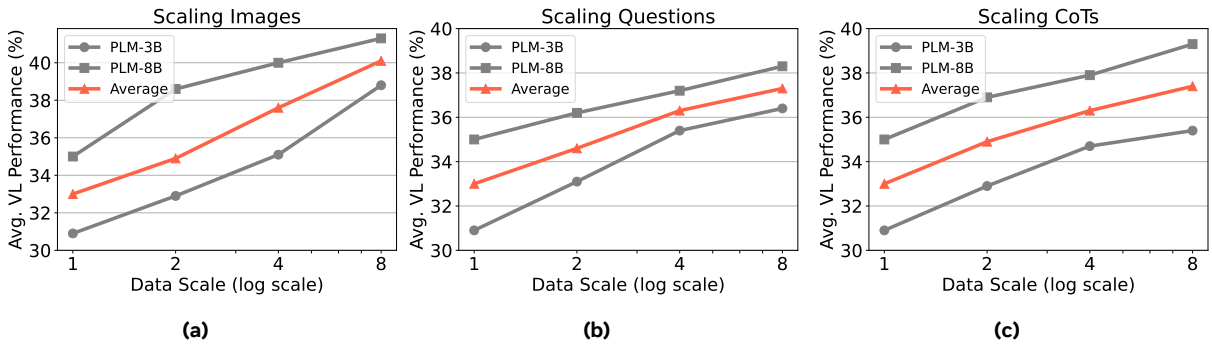


Figure 4 Impact of scaling diverse data axes in VL reasoning data. We train PLM-3B and PLM-8B on datasets of varying sizes (setup explained in §3.3). The results show that the reasoning performance consistently improves as we scale each data axis: (a) images, (b) synthetic questions per image, and (c) CoTs per (image, question) pair.

Statistic	Number (in K)
Total instances	2480
Number VL instances	1440
Number Text-Only instances	1040
Number Unique images	28
Number Unique questions	350
Avg. question length (words)	0.057
Avg. CoT length (words)	0.601

(a)

KeyWord	Category	Number (in K)
`Identify`	Perception	1950
`Recall`	Perception	1127
`Provided`	Comprehension	1040
`First`	Planning	2388
`Wait`	Reflection	32

(b)

Figure 5 HoneyBee statistics. (a) We present key statistics of the HoneyBee dataset. Notably, HoneyBee is a large-scale dataset with 2.5M instances with 350K unique questions. (b) We also provide statistics on several keywords in the HoneyBee CoTs that elicit diverse reasoning behaviors, such as perceptual understanding, question comprehension, planning, and reflection.

address this, we generate 4 CoTs per new question and use majority voting (an answer occurring three or more times) as a proxy for the final answer [50]. We then retain the CoTs for which the predicted answer matches the proxy final answer in the *scaling questions* scenario, leading to roughly 1M instances. In parallel, we generate image captions for all images in the source datasets. Following the *caption and solve* strategy, we combine the (image, question, solution CoT) tuples from the scaling CoTs and questions pipelines with the image captions to construct (image, question, (image caption, solution CoT)) tuples. This process results in the construction of the VL subset of the HoneyBee dataset, consisting of **1.5M** instances. We then merge this subset with the text-only reasoning data, consisting of **1M** instances, to obtain a high-quality, large-scale final HoneyBee dataset of size **2.5M**.

Dataset Statistics. We present the dataset statistics in Figure 5a. Beyond the total dataset size, we highlight that HoneyBee contains 28K unique images and 350K unique questions. The average length of CoTs (image caption and solution CoT combined) is approximately 600 words (780 tokens). Following [15], we analyze the occurrence of several keywords in our CoTs that are crucial for diverse reasoning behaviors, including visual understanding (perception), question comprehension, planning, and reflection. This suggests that the reasoning CoTs in our dataset encourage VL reasoners to elicit more complex reasoning (such as reflection) behaviors than existing VL reasoning datasets.

5.4 Training VL Reasoners with HoneyBee

HoneyBee elicits strong reasoning. Here, we assess the performance of PLMs of different sizes (1B, 3B) trained on the entire 2.5M HoneyBee dataset. We also compare them against several high-performing VLMs capable of generating CoTs to solve VL reasoning tasks. Results on several evaluation datasets are presented in Table

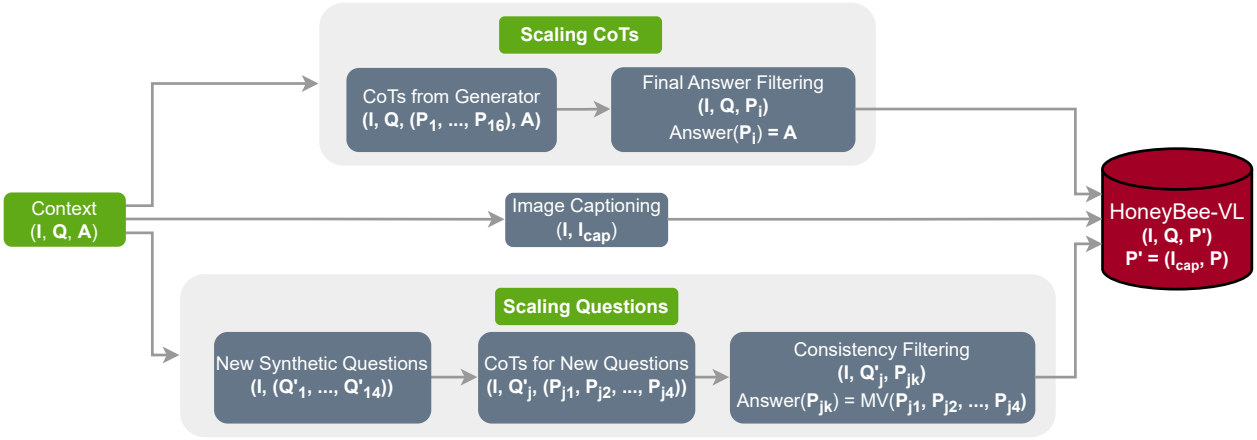


Figure 6 HoneyBee-VL scaling pipeline. We present the pipeline to scale the number CoTs per (real image, question) pair (**top**), and scale the number of questions per real image (**bottom**) from the context datasets. We denote majority voting over the several predicted answers as MV. Ultimately, this pipeline leads to the creation of 1.5M VL reasoning instances.

1. We find that VL reasoners trained with HoneyBee achieve the highest average accuracy across all model size categories. In particular, PLM-HoneyBee-1B achieves a relative performance improvement of 28pp over InternVL-3-1B-Instruct. Furthermore, PLM-HoneyBee-3B and PLM-HoneyBee-8B achieve relative gains of 8.4pp and 2.7pp over Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-8B-Instruct, respectively. Moreover, we perform a fine-grained comparison between the base PLM-3B and PLM-HoneyBee-3B across diverse difficulty levels in the `MathVision` dataset (Figure 7). We find that HoneyBee improves performance across all difficulty levels, reaching up to 100pp relative gains for level 2. This highlights that HoneyBee can be used to enhance reasoning capabilities at various difficulty levels. Overall, we show that our high-quality curated data can outperform existing SOTA methods that provide little public information about their reasoning CoT data. In addition, we observe that models trained on HoneyBee generalize well and achieve best-in-class performance on evaluation datasets that were unseen during curation, including robust reasoning in `DynaMath`, logical reasoning `LogicVista`, perception-centric `HallusionBench`, and STEM-reasoning `GPQA`.

HoneyBee data scaling. To understand the data scaling behavior with HoneyBee, we train PLM models on various subsets (50K, 250K, 2.5M) of the data. Results averaged across the five VL evaluation datasets are shown in Figure 1a. We observe that the performance of HoneyBee-trained PLMs, across all sizes, continues to improve as the dataset size increases. In fact, we find that performance has not saturated even at the 2.5M scale. This suggests that, with sufficient training budget, one could further scale the size of HoneyBee to achieve additional performance improvements. We show the scaling results across individual datasets in Appendix Figure 10. Further, we compare the PLM-3B trained with the HoneyBee data and existing CoT VL reasoning datasets in Figure 1b, and show massive relative gains upto 39pp across the five VL reasoning datasets. Apart from this, we use a small subset of our data to finetune the teacher model itself, reminiscent of self-improvement [55], and show that HoneyBee data can be used to achieve reasoning performance from the generator model too (Appendix C).

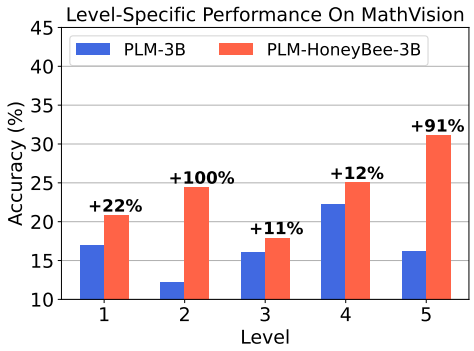


Figure 7 PLM-3B trained on HoneyBee outperforms the base model across all `MathVision` difficulty levels.

RL training of HoneyBee model. Supervised finetuning with CoT data serves as a warm start for RL training, enabling further improvements in reasoning capabilities [36, 15]. Thus, we perform RL training using the

Algorithm	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
OpenVLThinker-v1.2-3B [15]	42.3	34.1	63.9	25.0	**28.7**	59.8
PLM-HoneyBee-3B	44.3	42.8	61.2	**29.9**	28.4	59.3
PLM-HoneyBee-3B-GRPO	**46.2**	**43.5**	**64.9**	28.4	28.3	**65.8**

Table 6 RL training on top of HoneyBee trained VLM. We present results for training PLM-3B with HoneyBee data using supervised finetuning, followed by a round of RL training with the GRPO algorithm on a verifiable VL dataset. Furthermore, we compare its performance to that of a state-of-the-art RL-tuned VL reasoner, OpenVLThinker-v1.2-3B, which has a similar model capacity.

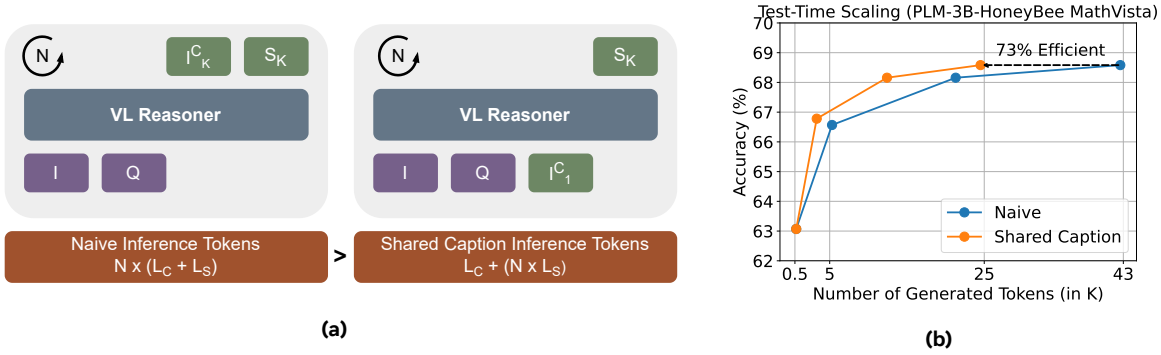


Figure 8 Shared caption decoding for efficient test-time scaling (TTS). (a) We illustrate the naive approach to TTS in VL reasoning and compare it to the proposed decoding strategy. After training with the HoneyBee dataset, we show that the number of generated tokens can be significantly reduced by reusing the caption generated for a given image in subsequent problem-solving attempts. We note that I_1^C is the image caption generated in the first attempt of solution generation. (b) We present the accuracy trend as a function of the number of generated tokens (number of solution attempts up till 64) for PLM-3B trained with HoneyBee on the downstream MathVista dataset.

GRPO algorithm [53] on top of the PLM-HoneyBee-3B model with VL verifiable data [37]. We present results on diverse VL reasoning tasks and compare them to those of a supervised and RL-tuned VL reasoner, OpenVLThinker-v1.2-3B, in Table 6. We find that RL training on top of HoneyBee-SFT models outperforms OpenVLThinker-v1.2-3B, with a 9.2pp relative gain in average accuracy. In particular, the RL-trained model achieves substantial improvements on the MathVision and We-Math datasets compared to the SFT model, with relative margins of 27.5pp and 10.0pp, respectively.

5.5 Efficient test-time scaling with Shared captions

We observe that each CoT in the HoneyBee dataset has two parts: an understanding component (I^C, image caption tokens) and a problem-solving component (S, solution tokens), so $C = [I^C; S]$. In test-time scaling (TTS) methods like self-consistency [68], we generate $N > 1$ CoTs for a reasoning problem (I, Q) and use majority voting over answers. The naive TTS approach generates the full CoT N times. Instead, we propose *shared captions*: generate the full CoT once as (I_1^C, S_1), then reuse I_1^C as context for subsequent solution generations S_K (Figure 8a). This reduces number of generated tokens and, since inference FLOPs scale with token count [25], improves inference efficiency. We empirically test this with PLM-3B trained on HoneyBee using MathVista (Figure 8b), generating $N = 64$ solutions at temperature 0.7 per (image, question) pair. The naive approach produces 42.6K tokens (671 per attempt), while *shared captions* achieves similar performance with only 24.5K tokens (280 captioning, 390 problem-solving per attempt), a 73% reduction in tokens and FLOPs. Thus, HoneyBee enables inference-time efficiency and strong VL reasoning.

5.6 Correlation Analysis

For our data curation, we train PLM-3B and PLM-8B and evaluate them on five VL reasoning tasks for each experiment to make robust decisions that genuinely improve reasoning performance across multiple model

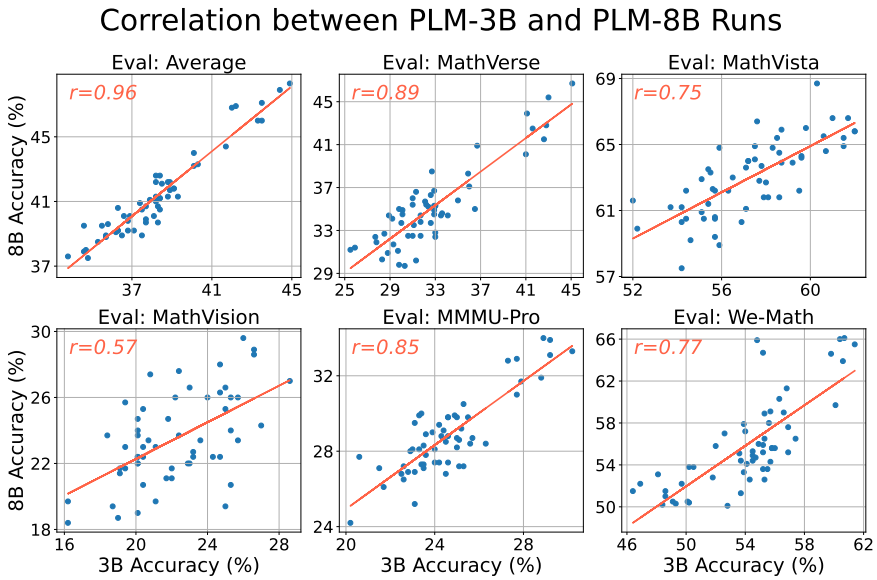


Figure 9 Model Correlation Analysis. We present the correlation between the performance of PLM-8B and PLM-3B, each trained with various data distributions throughout this work, across diverse downstream evaluation tasks. A higher correlation implies greater predictability between model performances; that is, a data curation method that is optimal for a 3B VLM is more likely to be optimal for an 8B VLM.

scales and evaluations. Here, we aim to understand the correlation between the performance achieved by PLM-8B and PLM-3B on average (and individual dataset) accuracy across all experiments conducted in this work (Figure 9), and observe a strong correlation in average accuracy. This indicates that data curation methods have similar impacts on average accuracy across different model scales. Similar to [41] for LLM pretraining, a broader implication of this analysis is that future research can perform optimal data selection strategies with a small VLM, and it is likely to be optimal for a larger VLM.

6 Related Work

Chain-of-Thought Reasoning Datasets. Prior work [70, 28] has shown that large language models (LLMs) can solve diverse reasoning tasks (e.g., math and logic) by generating chain-of-thoughts (CoTs) before providing their final answers at inference. This approach allows models to utilize additional computation at test time and further improve performance by aggregating decisions across several reasoning CoTs using methods such as self-consistency [68]. Notably, STaR [77] demonstrates that an LLM can be prompted to generate CoTs for novel questions, which can then be filtered based on final answer correctness. This synthetic process removes the need for expensive and time-consuming CoT collection from human annotators. Since then, several datasets [30, 46, 32, 61, 60, 6] have been proposed that bootstrap math CoT reasoning traces from strong LLMs [1, 20]. In particular, OpenThoughts [19] is a state-of-the-art CoT reasoning dataset, constructed through comprehensive ablation across several key data design decisions and ultimately scaled to 1.2M reasoning examples, predominantly centered on math problem-solving. In comparison, our work focuses on the more complex scenario of math reasoning in visual contexts, where vision-language (VL) models must integrate information from multiple modalities, image and text, instead of just text. Specifically, we create HoneyBee, a high-quality VL CoT dataset that elucidates the data design process, including data sources, filtering strategies, and scaling properties at multiple model scales (e.g., 3B and 8B). Furthermore, we scale our dataset to 5M VL instances and observe consistent improvements with increased data size.

VL Reasoning Datasets. Math problem-solving serves as a critical testbed for assessing the reasoning capabilities of VL models, as it requires them to develop a deep understanding and interpretation skills across diverse visual categories such as geometry, functions, and charts [40, 79, 66]. To enable VL reasoning, prior work has proposed reasoning datasets, such as MathV360K [54], which selects and creates high-quality questions

from a collection of existing image-question datasets. However, this dataset does not provide any CoT data, making it unsuitable for eliciting complex reasoning behaviors in VL models. Other datasets, such as MAVIS [80], focus on creating novel questions and CoTs for plane geometry and functions using an automatic data engine. However, such synthetic data is unsuitable for achieving strong performance on diverse real-world visual scenarios, such as tabular understanding, IQ tests, science question-answering, and math reasoning on natural scenes. More recently, datasets such as `LLaVA-CoT` [73] and `R1-OneVision` [74] have sourced image-question data from diverse sources, including the web and textbooks, followed by CoT generation and multiple rounds of data curation from capable teacher VL models [23, 56]. However, it remains unclear whether the benefits of these datasets are attributable to differences in data sources, choice of teacher models, or specific data filtering strategies. In this work, we take a comprehensive approach by re-annotating the CoTs in each dataset with a given teacher model to understand the impact of diverse data sources, and we also study the effect of mixing the contexts in these datasets. Further, we curate a list of interventions (e.g., filtering, augmentation, and replacement) and test their impact on enhancing the performance of VL models of capacity. Subsequently, our approach examines the effect of scaling across several data dimensions, such as unique images, questions, and CoTs, which ultimately enables the development of HoneyBee, one of the largest VL reasoning datasets to date.

VL Data Curation. Reasoning in visual contexts is a fundamental capability of modern AI systems [49, 57, 45, 58, 2, 12], enabling a wide range of real-world applications such as visual data analysis and scientific discovery. However, the training data for these systems are not publicly available, making it challenging to study the underlying science behind the development of state-of-the-art VL reasoning CoT datasets. In this work, we aim to investigate the science behind curating high-quality, large-scale VL reasoning datasets. To this end, we curate a list of several data interventions designed to improve the perception and reasoning capabilities of these models. Prior work [33] has shown that visual perturbations (e.g., rotation, distractors) can enhance perceptual robustness and lead to better mathematical reasoning. In addition, [79] highlights that VL models often rely primarily on the information about the visual content provided in the question, rather than truly integrating visual information. Further, [76] argues that augmenting the original questions to have 10 options instead of 4 reduces VL model performance on general reasoning tasks. [11] also demonstrates that VL reasoners benefit from exposure to unimodal (text) reasoning data. Other work [9] shows that perception-enhanced reasoning chains improve the ability of VL models to play games such as Shisen-Sho. However, all these potential data interventions use different training paradigms (e.g., instruction-tuning, RL), model families and sizes, and underlying evaluations. This makes it difficult to anticipate which strategies will lead to reliable improvements in VL reasoning. To address this, we perform a comprehensive set of experiments in which several datasets are compared against each other using a fixed training algorithm, student models and their sizes, and a battery of VL reasoning evaluation datasets. Our findings help in creating the best VL reasoning dataset, HoneyBee, and show that model performance improves with scale, along with efficiency benefits for test-time scaling.

7 Conclusion

In this report, we present HoneyBee, a high-quality, large-scale VL reasoning dataset with chain-of-thoughts (CoTs). Future work can assess the impact of our insights on general-purpose data curation for VL training, particularly for skills that go beyond reasoning, such as VQA. Moreover, we have focused only on data curation for single images, but it would be pertinent to extend this approach to reasoning over multiple images. Overall, our work lays a strong foundation for data research in VL reasoning.

References

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

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

[4] Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q Tran, and Mehran Kazemi. Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling. *arXiv preprint arXiv:2408.16737*, 2024.

[5] Sami Baral, Li Lucy, Ryan Knight, Alice Ng, Luca Soldaini, Neil T Heffernan, and Kyle Lo. Drawedumath: Evaluating vision language models with expert-annotated students' hand-drawn math images. *arXiv preprint arXiv:2501.14877*, 2025.

[6] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, Eric Chung, and Chris Alexiuk. Llama-nemotron: Efficient reasoning models, 2025. https://arxiv.org/abs/2505.00949.

[7] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

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

[9] Liang Chen, Hongcheng Gao, Tianyu Liu, Zhiqi Huang, Flood Sung, Xinyu Zhou, Yuxin Wu, and Baobao Chang. G1: Bootstrapping perception and reasoning abilities of vision-language model via reinforcement learning. *arXiv preprint arXiv:2505.13426*, 2025.

[10] Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time?(2023). *arXiv preprint arXiv:2307.09009*, 2023.

[11] Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning. *arXiv preprint arXiv:2506.04207*, 2025.

[12] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[13] Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad

Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. Perceptionlm: Open-access data and models for detailed visual understanding. *arXiv preprint arXiv:2504.13180*, 2025.

[14] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[15] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: Complex vision-language reasoning via iterative sft-rl cycles. *arXiv preprint arXiv:2503.17352*, 2025.

[16] Yifan Du, Zikang Liu, Yifan Li, Wayne Xin Zhao, Yuqi Huo, Bingning Wang, Weipeng Chen, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Virgo: A preliminary exploration on reproducing o1-like mllm. *arXiv preprint arXiv:2501.01904*, 2025.

[17] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

[18] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385, 2024.

[19] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

[20] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[21] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[23] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[24] Yiming Jia, Jiachen Li, Xiang Yue, Bo Li, Ping Nie, Kai Zou, and Wenhu Chen. Visualwebinstruct: Scaling up multimodal instruction data through web search. *arXiv preprint arXiv:2503.10582*, 2025.

[25] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[26] Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Peter Chen, et al. Big-bench extra hard. *arXiv preprint arXiv:2502.19187*, 2025.

[27] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[28] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[29] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[30] Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*, 2024.

[31] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina,

Amro Kamal Mohamed Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Joshua P Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander T Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-LM: In search of the next generation of training sets for language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. https://openreview.net/forum?id=CNWdWn47IE.

[32] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

[33] Yuting Li, Lai Wei, Kaipeng Zheng, Jingyuan Huang, Linghe Kong, Lichao Sun, and Weiran Huang. Vision matters: Simple visual perturbations can boost multimodal math reasoning. *arXiv preprint arXiv:2506.09736*, 2025.

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

[35] Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*, 2022.

[36] Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. *arXiv preprint arXiv:2506.13284*, 2025.

[37] lmmslab. lmms-lab/multimodal-open-r1-8k-verified · Datasets at Hugging Face — huggingface.co. https://huggingface.co/datasets/lmms-lab/multimodal-open-r1-8k-verified, 2025.

[38] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Intergps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

[39] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

[40] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[41] Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, et al. Datadecide: How to predict best pretraining data with small experiments. *arXiv preprint arXiv:2504.11393*, 2025.

[42] Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*, 2024.

[43] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[44] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[45] AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on*, 4(7):2025, 2025.

[46] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*, 2024.

[47] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[48] OpenAI. Openai o3 and o4-mini system card. https://api.semanticscholar.org/CorpusID:277857808.

[49] OpenAI. Gpt-4v(ision) system card. 2023. https://api.semanticscholar.org/CorpusID:263218031.

[50] Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane Yu. Self-consistency preference optimization. *arXiv preprint arXiv:2411.04109*, 2024.

[51] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

[52] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[53] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[54] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.

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

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

[57] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[58] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

[59] NovaSky Team. Sky-t1: Train your own o1 preview model within 450 usd. https://novasky-ai.github.io/posts/sky-t1, 2025.

[60] Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

[61] Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *Advances in Neural Information Processing Systems*, 37: 34737–34774, 2024.

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

[63] The vLLM Team. Llama 4 in vLLM — blog.vllm.ai. https://blog.vllm.ai/2025/04/05/llama4.html, 2025.

[64] Rohan Wadhawan, Hritik Bansal, Kai-Wei Chang, and Nanyun Peng. Contextual: Evaluating context-sensitive text-rich visual reasoning in large multimodal models. *arXiv preprint arXiv:2401.13311*, 2024.

[65] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025.

[66] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[67] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025.

[68] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[69] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[70] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[71] Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

[72] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *ArXiv*, abs/2305.10429, 2023.

[73] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024.

[74] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

[75] Yuqing Yang, Yan Ma, and Pengfei Liu. Weak-to-strong reasoning. *arXiv preprint arXiv:2407.13647*, 2024.

[76] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.

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

[78] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[79] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

[80] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.

[81] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[82] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *Advances in Neural Information Processing Systems*, 36:8958–8974, 2023.

[83] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*, 2024.

Appendix

A Additional Context Dataset Details and Results

We present additional details about the context datasets in Table 7. Specifically, we report the total number of examples originally present in each dataset. Subsequently, we perform decontamination by removing instances in which the image exactly matches any image in the validation dataset, as determined by the phash algorithm. For the context curation experiments, we randomly sample at most 50K examples from each dataset, as shown in the third column. Furthermore, we create a filtered version of these datasets based on final answer correctness, with the resulting sizes reported in the last column.

Context Sources	Total Examples (in K)	# Contaminated	# Context Curation (Original, in K)	# Context Curation (Filtered, in K)
ViRL	38.9	80	38.9	23.6
Math-LLaVA	338.7	856	50.0	29.0
R1-OneVision	154.6	508	50.0	22.5
ThinkLite-VL-Hard	11.0	129	11.0	4.0
LLaVA-CoT	98.6	336	50.0	27.4
MMK12	17.6	1	17.6	9.0

Table 7 Context source dataset statistics. We present the total number of examples, the number of instances found to be contaminated, and the number of instances originally used in context curation experiments compared to their filtered versions.

Next, we train PLM-3B and PLM-8B on both the original and filtered versions of the source datasets, where CoTs are generated by our generator model (Llama-4-Scout). We evaluate the trained models on five downstream VL tasks and present their average accuracy in Table 8. We find that ViRL and ThinkLite-VL-Hard benefit from final answer correctness filtering, while the original (unfiltered) datasets produce better reasoners. The best-performing version of each context source dataset is presented in the main Table 2.

Context Sources		Original	Filtering
ViRL	3B	38.1	38.8
	8B	40.1	41.3
	Avg.	39.1	**40.1**
Math-LLaVA	3B	36.3	36.2
	8B	39.2	39.1
	Avg.	**37.7**	37.6
R1-OneVision	3B	35.7	34.8
	8B	38.8	37.5
	Avg.	**37.3**	36.2
ThinkLite-VL-Hard	3B	34.7	34.6
	8B	38.0	39.5
	Avg.	36.4	**37.1**
LLaVA-CoT	3B	34.6	33.8
	8B	37.9	37.6
	Avg.	**36.3**	35.7
MMK12	3B	34.6	35.6
	8B	37.3	36.0
	Avg.	**36.0**	35.8

Table 8 Impact of filtering on final answer correctness. We present the results of training PLM-3B and PLM-8B on both the original and filtered versions of the context source datasets. Specifically, we report the average accuracy across the five evaluation datasets and bold the best-performing setup.

B Additional Data Intervention Details and Results

Here, we present additional implementation details regarding the diverse data interventions. Where relevant, we also highlight experimental results across various configurations.

Visual Perturbation. In this intervention, we apply one of three transformations *rotation*, *distractor concatenation*, *dominance-preserving mixup*} to every image in the dataset. In *rotation*, we rotate the image by any angle (in degrees) in this list `[0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, 345]`. In *distractor concatenation*, we treat any image from the other dataset as a distractor, and concatenate it with the original image in the `width` dimension to form a new image. In *dominance-preserving mixup*, we form a new image in the input space by mixing the original image with a distractor image using $\texttt{mixup image} = \alpha \times \texttt{original image} + (1-\alpha) \times \texttt{distractor image}$ where $\alpha \sim \mathcal{U}[0.8, 1.0)$.

We test two approaches for applying visual perturbation to the original data: (a) *replacement*, where a 50% of the original data is replaced with its perturbed version, and (b) *augmentation*, where the perturbed version of the original data is concatenated with the original data itself. For (b), we compare the performance of training VLMs on 50% of the original data versus training them on the combination of this data and its visually perturbed version, effectively doubling the amount of training data at no additional cost. The results are shown in Table 9. We find that augmenting the original data with its visually perturbed version leads to worse reasoning performance, which may be attributed to increased redundancy in the training data and the noise introduced by visual perturbations.

Augmentation Expts.	Models	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
50% Original Data	3B	35.1	28.7	52.7	23.0	25.0	46.1
	8B	40.0	32.6	61.5	24.3	28.4	53.2
	Avg.	**37.6**	30.6	57.1	23.7	26.7	49.7
50% Original Data + 50% Visual Perturbation	3B	34.6	28.3	52.6	19.1	24.5	48.5
	8B	39.2	31.9	62.0	23.4	27.4	51.4
	Avg.	36.9	30.1	57.3	21.2	26.0	50.0
50% Original Data + 50% Text-Rich Images	3B	34.8	28.7	55.4	16.1	25.4	48.2
	8B	39.4	32.9	59.5	22.7	28.1	53.7
	Avg.	37.1	30.8	57.5	19.4	26.8	51.0

Table 9 Data intervention augmentation results.

Text-Rich Images. In this intervention, we render the original image and question from the dataset into a new in the raw space on randomly sampled background colors, text fonts and colors. We use functions from the `PIL` python library to achieve this task. In particular, we provide our rough implementation in Listing E. Similar to visual perturbations, the *text-rich images* intervention can be used in two ways: (a) *replacement* on 50% of original data, and (b) *augmentation*. We highlight the results for *augmentation* strategy in Table 9. In particular, we find that augmenting the data with text-rich images does not improve the reasoning performance over the half the original data (row 1 vs row 3).

Perceptual Redundancy. This strategy filters the instances in the original data if the blind generator model can solve the question correctly i.e., without access to the image. We observe that this strategy removes 42% of the instances in the data curation experiments.

Shallow Perception. This strategy is a stricter version of the perceptual redundancy where the instances that can be solved correctly using a blind generator model with access to image captions are filtered. In our experiments, we observe that this strategy removes 55% of the data.

Caption and Solve. In this strategy, the chain-of-thought (CoT) is augmented to first generate an image caption, providing auxiliary perceptual signals before addressing the given question. Specifically, we explore three approaches for creating such data using our generator model: (a) synthesizing the image caption and

Caption and Solve Variants	Models	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math
I → C (I, Q) → S	3B	39.7	35.7	56.6	24.7	26.7	55.0
	8B	43.0	38.3	63.2	26.3	30.4	56.9
	Avg.	**41.4**	37.0	59.9	25.5	28.6	56.0
(I, Q) → (C, S)	3B	37.8	32.5	55.1	22.4	25.8	53.4
	8B	42.3	38.4	60.4	27.6	29.0	55.8
	Avg.	40.0	35.5	57.7	25.0	27.4	54.6
I → C (I, Q, C) → S	3B	36.6	31.2	52.3	23.1	24.4	52.1
	8B	38.9	30.2	60.5	22.0	29.4	52.6
	Avg.	37.8	30.7	56.4	22.5	26.9	52.3

Table 10 Caption and Solve variants. We present the accuracy of models trained with different data generation variants for the *caption and solve* intervention. Here, I → C means that the teacher model takes image I as input and outputs the caption C. Furthermore, (I, Q) → S indicates that the teacher model takes the (image, question) pair as input and outputs the solution S. We note that the student VL reasoner model is always trained to generate the caption C and solution S jointly, conditioned on the image and question pair: (I, Q) → (C, S).

problem-solving steps independently i.e., $I_j^{cap} = \mathcal{G}(I_j)$ and $C_j = \mathcal{G}(I_j, Q_j)$, (b) generating the image caption first, followed by problem-solving $(I_j^{cap}, C_j) = \mathcal{G}(I_j, Q_j)$, and (c) generating the image caption and then providing it as additional context for problem-solving i.e., $C_j = \mathcal{G}(I_j, Q_j, I_j^{cap})$. Ultimately, the VLMs are trained in an identical fashion across all three strategies; that is, the input consists of (image, question), and the CoT comprises (caption, solution). We present the results from these three strategies in Table 10. We observe that the first strategy produces CoTs that train the strongest VL reasoners. This approach is also the most efficient in terms of scalable data generation, as the caption and solution can be generated in parallel and later merged for training VL reasoners.

Text-Only Reasoning. In this strategy, we augment the original VL reasoning data with the text-only reasoning data. In particular, we choose a state-of-the-art reasoning data, `OpenThoughts3` [19]. The CoTs in the original dataset are collected the QwQ-32B reasoning model.[9] To ensure consistency between the generator models, we re-annotate the entire 1.04 million samples with Llama-4-Scout.

Increased Distractors. In this strategy, we re-write the dataset to increase the number of distractors (Listing E). In our experiments, we consider the *replacement* strategy with this intervention on 50% of original data.

Length. In this strategy, we bias our dataset to elicit long CoTs. Specifically, we compute the median CoT length and divide the dataset into two equal halves i.e., instances with CoTs lesser than the median and the ones more than the median.

Difficulty. In this strategy, we classify each instance in the dataset according to its difficulty level. In our dataset of size 50K, the distribution of difficulty levels is `{Level 2: 15.1K, Level 1.5: 14.4K, Level 1: 11.8K, Level 4: 3.6K, Level 3: 3.5K, Level 4+: 0.5K}`. To increase the impact of harder examples, we select 3500 samples from each difficulty level and train on this uniform-difficulty dataset.

C Self-Improvement

Here, we aim to study the impact of training the teacher model, Llama-4-Scout (109A17B MoE), on the self-generated reasoning data, HoneyBee. Specifically, we take 50K subset of this data and finetune this model using low-rank adaptation (LoRA) with the `Llama-factory` library.[10] We present the results across downstream evaluation datasets in Table 11. We find that parameter-efficient finetuning of Llama-4 on the HoneyBee dataset leads to a relative gain of 3.7pp over the base model. Importantly, we observe

[9] https://qwenlm.github.io/blog/qwq-32b/
[10] https://github.com/hiyouga/LLaMA-Factory/pull/7611

	Average	MathVerse	MathVista	MathVision	MMMU-Pro	We-Math	MATH500	GPQA
Llama-4-Scout	59.3	53.9	69.3	32.6	51.2	72.9	81.2	54.0
Llama-4-Scout (HoneyBee)	**61.5**	**54.3**	**70.9**	**37.5**	**51.7**	72.8	**83.0**	**60.6**

Table 11 Self-improvement results. We train the generator model, Llama-4-Scout, on a subset of the HoneyBee dataset and report its accuracy on diverse evaluation datasets.

improvements across most of the evaluation datasets indicating the strong self-improvement capabilities. Future work can focus on training much stronger models on using our high-quality and large-scale dataset.

D Additional Training and Evaluation Setup

PLM setup. In this work, we predominantly train Perception Language Models (PLMs) [13] for our data curation and large-scale experiments. Specifically, the PLMs use a Llama-3.2 LLM backbone that processes visual input via a perception encoder [7]. We train PLMs using their publicly available codebase, starting with the stage-3 configurations.[11] In particular, we perform full finetuning of all parameters of the PLMs including the vision encoder and the LLM backbone. Due to the sheer number of experiments, we use the default peak learning rate (LR) of 4e−5 for PLM-1B and PLM-3B, and 1e−5 for PLM-8B. The warmup ratio is set to 0.1 followed by cosine decay upto 10% of the peak learning rate. Furthermore, we train our models for 5 epochs across all experiments. In our data curation experiments, we set a global batch size of 2048 for PLM-1B and PLM-3B, and 1024 for PLM-8B, while we double the global batch sizes for the large-scale runs. All experiments are performed on multiple nodes i.e., either 8 or 16 A100 (80GB) nodes.

RL Training. In Table 6, we present a round of RL training using the GRPO algorithm. Specifically, we use the default implementation from the TRL library.[12] Since this work is focused on supervised finetuning, we conduct lightweight GRPO training (1 epoch) to demonstrate its effectiveness as a cold start for RL. In particular, we use a verifiable multimodal dataset.[13] We set the number of generations per prompt in the GRPO algorithm to four. Further, we use the peak learning rate of 1e−6 works well for PLM-HoneyBee SFT models in the TRL configuration.

Evaluation. To ensure consistent prompting across all benchmarks, we query our model with '`Your final answer MUST BE put in` $\boxed{boxed}$.' Furthermore, we score our models using the rule-based verifier `mathruler` for scalability and fast iteration following [15].[14] We notice that the original `MathVerse` free-form questions often contain numerics followed by units (e.g., celsius, cm), which makes rule-based verification more challenging. To address this, we use an LLM to extract only the final answer values and consider them as the ground-truth answers.

[11] https://github.com/facebookresearch/perception_models
[12] https://github.com/huggingface/trl/blob/main/examples/scripts/grpo_vlm.py
[13] lmms-lab/multimodal-open-r1-8k-verified
[14] https://github.com/hiyouga/MathRuler, https://github.com/yihedeng9/OpenVLThinker

E Prompts

CoT generation prompt for a given (image, question) pair.

```
Think about the reasoning process required to solve the question in detail based
on the given image and then provide the answer. The reasoning process should be
enclosed within <think> and </think> tags followed by the summary of the answer.

In your reasoning, incorporate the following elements:

- Planning: Systematically outline the steps before executing them.
- Evaluation: Critically assess each intermediate step.
- Reflection: Reconsider and refine your approach as needed.
- Exploration: Consider alternative solutions.

If the question is multiple-choice, present the final answer as the option letter
A, B, C, ... F.

Enclose the final answer in \\boxed{}.

Question: <Question>
```

Prompt to generate new question for a given (image, question) pair.

```
Help the user create a new math problem that inspired by the original one related
to the provided image.

- Make sure the new problem is logical and can be solved.
- The new question should be directly based on the image, not an independent
question.
- Do not solve the original question.
- Conclude your response with "New Question: ". If necessary, please include
options or choices in the new question.
- Do not provide the answer in your new question.

Original Question: <Question>
```

Prompt to generate a detailed caption for a given image.

```
Provide a detailed caption of the given image.

Caption:
```

Prompt to rewrite the original (question, CoT) pair with distractors.

```
Please rewrite the question shown in the image to include a total of 10 options.
- Rewrite the given question to include a total of 10 options. If the original
question does not have any options, create and add 10 options to it.
- If the original question has options then the original options should be
shuffled (e.g., content of option A goes to option D etc.)
- Rewrite the predicted answer in its entirety, maintaining the same step-by-step
structure and content as the original answer.
- Ensure that the final answer should be in the format of \\boxed{Option Letter},
 where "Option Letter" corresponds to the correct choice.
- Do not attempt to solve the question at any time, just focus on rewriting.

Question: <Question>

```

```
Answer: <Answer>

Respond with "Rewrite Question: " and "Rewrite Answer: ".
```

Code snippet to create text-rich images.

```
from PIL import Image, ImageDraw, ImageFont

# Text wrapping function
def wrap_text(text, font, max_width):
    """Wrap text to fit within max_width pixels"""
    words = text.split()
    lines = []
    current_line = []

    for word in words:
        # Test if adding this word would exceed max width
        test_line = ' '.join(current_line + [word])
        bbox = draw.textbbox((0, 0), test_line, font=font)
        text_width = bbox[2] - bbox[0]

        if text_width <= max_width:
            current_line.append(word)
        else:
            if current_line:
                lines.append(' '.join(current_line))
                current_line = [word]
            else:
                # Single word is too long, force it
                lines.append(word)

    if current_line:
        lines.append(' '.join(current_line))

    return lines


def get_text_rich_image(image_path, question, save_path, background_colors, text_colors, text_fonts):

    """
    background_colors, text_colors: list of RGB codes
    text_fonts: list of .ttf files
    """

    ## blank canvas
    img_width = random.choice(img_widths)
    img_height = 512 * 4
    background_color = random.choice(background_colors)

    # Create a new image with the background color
    img = Image.new('RGB', (img_width, img_height), color=background_color)
    draw = ImageDraw.Draw(img)

    ## embed text
    text_color = random.choice(text_colors)
    font = ImageFont.truetype(random.choice(text_fonts), 16)  # Linux

    margin = 20
    max_text_width = img_width - (2 * margin)
    line_spacing = 5

    # Wrap the text
    embedded_question = question.replace("\n", " ")
```

```
        wrapped_lines = wrap_text(embedded_question, font, max_text_width)

        # Calculate line height
        sample_bbox = draw.textbbox((0, 0), "Sample", font=font)
        line_height = sample_bbox[3] - sample_bbox[1]

        # Draw the wrapped text
        y_position = margin
        for line in wrapped_lines:
            draw.text((margin, y_position), line, fill=text_color, font=font)
            y_position += line_height + line_spacing

        ## embed image
        overlay_image = Image.open(image_path)

        original_overlay_img_width = overlay_image.width
        if original_overlay_img_width > int(0.8 * img_width):
            overlay_width = int(0.8 * img_width)
        else:
            overlay_width = original_overlay_img_width
        original_width, original_height = overlay_image.size

        aspect_ratio = original_height / original_width
        overlay_height = int(overlay_width * aspect_ratio)

        overlay_img_resized = overlay_image.resize((overlay_width, overlay_height),
        Image.Resampling.LANCZOS)

        if overlay_img_resized.mode == 'RGBA':
            img.paste(overlay_img_resized, (int(0.1 * img_width), y_position + margin)
            , overlay_img_resized)
        else:
            img.paste(overlay_img_resized, (int(0.1 * img_width), y_position + margin)
            )

        height = y_position + overlay_height + (2 * margin)

        ## save image
        img = img.crop((0, 0, img_width, height))
        img.save(save_path)
```

F Qualitative Examples

We show several qualitative examples of newly generated questions for a given image in Figure 11 and Figure 12. These examples highlight that the synthetic questions can be quite diverse and reasonable for training VL reasoners. Furthermore, we present a few qualitative examples from the HoneyBee dataset in Figure 13 and Figure 14. These examples demonstrate that the CoTs in our dataset are formed by concatenating the captions and problem solutions in the raw text space.

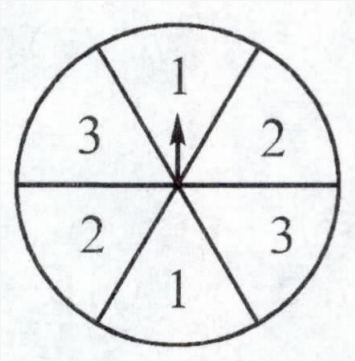
Train schedule			
Rollercoaster Land	9:00 A.M.	2:45 P.M.	9:30 P.M.
Chickenville	9:30 A.M.	3:15 P.M.	10:00 P.M.
City of Green Trees	10:15 A.M.	4:00 P.M.	10:45 P.M.
Floral Gardens	10:45 A.M.	4:30 P.M.	11:15 P.M.
Comfy Pillows Resort	11:15 A.M.	5:00 P.M.	11:45 P.M.

Synthetic Question 1: What is the travel time from Rollercoaster Land to Chickenville?
Choices:
(A) 15 minutes
(B) 30 minutes
(C) 45 minutes
(D) 1 hour

Synthetic Question 2: What is the travel time from City of Green Trees to Floral Gardens?
Choices:
(A) 30 minutes
(B) 45 minutes
(C) 1 hour
(D) 15 minutes

Synthetic Question 3: What is the most common time interval between consecutive train stops according to the schedule?
Choices:
(A) 15 minutes
(B) 30 minutes
(C) 45 minutes
(D) 1 hour

Figure 11 Qualitative example to show diverse and reasonable synthetic images for a given image.



Synthetic Question 1: What is the probability that the pointer lands on a number greater than 1 when the spinner is spun once?

Synthetic Question 2: If the spinner is spun twice, what is the probability that the pointer lands on a number greater than 1 on both spins?

Synthetic Question 3: What is the probability that the pointer lands on a sector labeled with the number 2 when the spinner is spun once?

Figure 12 Qualitative example to show diverse and reasonable synthetic images for a given image.

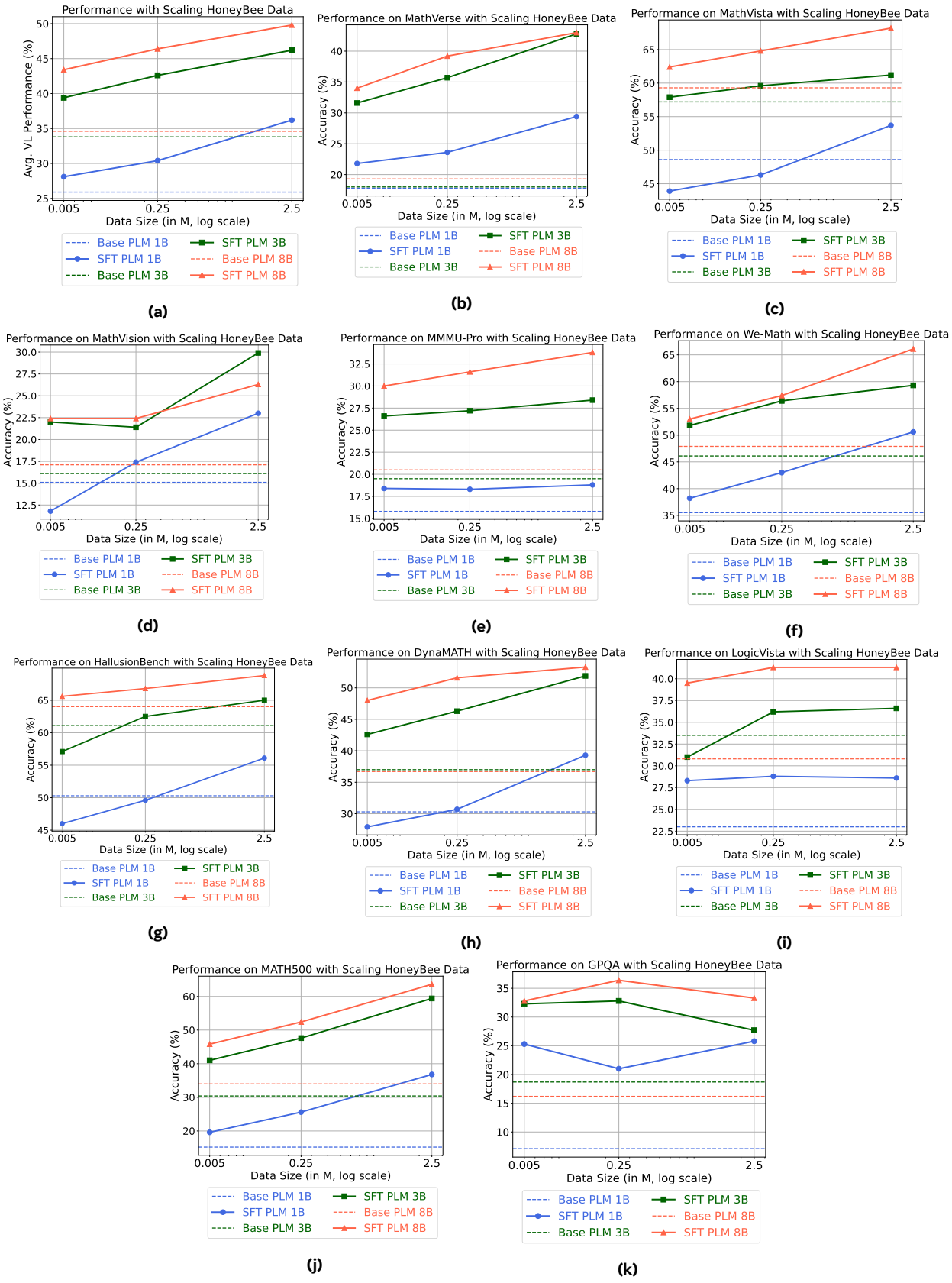


Figure 10 Scaling trends across individual evaluation datasets. We train PLMs (1B-8B) on diverse subsets of the HoneyBee data and evaluate them on individual datasets. The first plot (first row, first column) is the average performance across the ten evaluation datasets.

Time Cody spent on homework	
Day	Minutes
Sunday	38
Monday	38
Tuesday	35
Wednesday	34
Thursday	31

Question: What is the total time Cody spent on homework from Sunday to Tuesday?
A) 105 minutes
B) 111 minutes
C) 106 minutes
D) 108 minutes.

Chain-of-Thought:
<caption> The image presents a table illustrating the time Cody spent on homework each day, with data for five days.
Table Structure:
The table consists of two columns and six rows.
The first row is a header row, with the left column labeled D̈ayänd the right column labeled M̈inutes:
The remaining five rows represent individual days of the week, from Sunday to Thursday.
Data:
Sunday: 38 minutes
Monday: 38 minutes
Tuesday: 35 minutes
Wednesday: 34 minutes
Thursday: 31 minutes
In summary, the table provides a clear and concise overview of Cody's daily homework time, allowing for easy comparison and analysis of his study habits over the course of a week. </caption>
<think>
Step 1: Identify the relevant data
To find the total time Cody spent on homework from Sunday to Tuesday, we need to identify the time spent on each of these days. According to the table, Cody spent 38 minutes on Sunday, 38 minutes on Monday, and 35 minutes on Tuesday.
2: Add up the time spent on each day
Now, we add up the time spent on Sunday, Monday, and Tuesday.

$$\text{total\_time} = 38 + 38 + 35$$

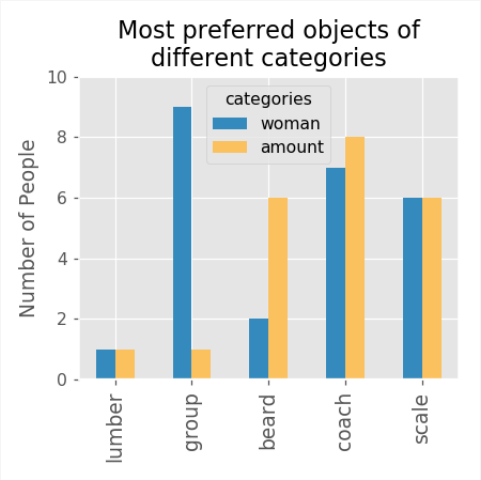
3: Calculate the total time
Perform the addition to find the total time.

$$\text{total\_time} = 38 + 38 + 35 = 111 \text{ minutes}$$

4: Evaluate the answer choices
Compare the calculated total time to the given answer choices to select the correct one. The choices are A: 105 minutes, B: 111 minutes, C: 106 minutes, D: 108 minutes.
5: Select the correct answer
Based on the calculation, the total time Cody spent on homework from Sunday to Tuesday is 111 minutes, which matches option B. The steps taken to solve this problem involved directly adding up the time Cody spent on homework from Sunday to Tuesday. The process was straightforward, and the calculation was simple. There were no alternative solutions needed since the question only required a basic arithmetic operation. The reflection on the steps confirms that the approach was correct and efficient for this type of problem.
</think>

$$\boxed{B}$$

Figure 13 Qualitative example from HoneyBee dataset.



Question: How many total people preferred the object coach across all the categories?

Chain-of-Thought:

<caption> The image presents a bar graph illustrating the most preferred objects across different categories, with two distinct groups: "woman" and "amount." The graph features five categories on the x-axis: lumber, group, beard, coach, and scale.
Key Observations:
*The y-axis represents the number of people, ranging from 0 to 10.
*The graph displays two sets of bars for each category, one in blue (representing "woman") and the other in orange (representing "amount").
*Notably, the "group" category has the highest value for "woman," while the "coach" category has the highest value for "amount."
Category-Specific Data:
***Lumber:** *Woman: 1 *Amount: 1
***Group:** *Woman: 9 *Amount: 1
***Beard:** *Woman: 2 *Amount: 6
***Coach:** *Woman: 7 *Amount: 8
***Scale:** *Woman: 6 *Amount: 6
Conclusion: The graph provides a clear visual representation of the preferred objects across different categories, allowing for easy comparison between the "woman" and "amount" groups. </caption>
<think>
Step 1: Identify the relevant data The image shows a bar graph with different categories on the x-axis and the number of people on the y-axis. The categories are lumber, group, beard, coach, and scale. We are interested in the category "coach".
To solve this question, I will first identify the values associated with the "coach" category for both woman and amount. Then, I will add these values to find the total number of people who preferred the object "coach" across all categories.
2: Extract the values for the coach category For the "coach" category, the value for woman is 7 and the value for amount is 8.
I will now evaluate if these values are correctly identified and if they align with the question asked. The question is about the total number of people who preferred the object "coach", so I need to ensure I'm considering both woman and amount as they represent different aspects of preference.
3: Calculate the total number of people To find the total, I add the number of woman who preferred coach (7) and the amount of people who preferred coach (8).

$$\text{total\_people} = 7 + 8 = 15$$

I will reflect on my approach to ensure it's correct. The approach involves directly adding the values for woman and amount in the coach category. This seems straightforward and aligns with basic principles of data analysis. There are no obvious alternative solutions since the question directly asks for a sum.
4: Provide the final answer Given the calculation, the total number of people who preferred the object "coach" across all categories is 15. </think>

$$\boxed{15}$$

Figure 14 Qualitative example from HoneyBee dataset.