Can Vision-Language Models Solve Visual Math Equations?

Anonymous ACL submission

Abstract

Despite strong performance in visual understanding and language-based reasoning, Vision-Language Models (VLMs) struggle with tasks requiring integrated perception and symbolic computation. We study this limitation through *visual equation solving*, where mathematical equations are embedded in images, variables are represented by object icons, and coefficients must be inferred by counting. While VLMs perform well on textual equations, they fail on visually grounded counterparts. To understand this gap, we decompose the task into coefficient counting and variable recognition, and find that counting is the primary bottleneck, even when recognition is accurate. We also observe that composing recognition and reasoning introduces additional errors, highlighting challenges in multi-step visual reasoning. Finally, as equation complexity increases, symbolic reasoning itself becomes a limiting factor. These findings reveal key weaknesses in current VLMs and point toward future improvements in visually grounded mathematical reasoning.[1]

1 Introduction

Vision-Language Models (VLMs) have become the dominant architecture for multimodal learning, powering applications such as visual question answering (Ghosal et al., 2023), image captioning (Yang et al., 2023), and multimodal reasoning (Li et al., 2024b). As agentic AI systems gain traction, VLMs are increasingly expected to function as general-purpose perception-and-reasoning modules for intelligent agents (Li et al., 2024b,a). While recent models demonstrate strong capabilities in both visual understanding and language-based reasoning, truly agentic behavior demands deeper integration, particularly in tasks involving grounded mathematical reasoning (Shi et al., 2024).

In this work, we investigate this integration through the lens of a seemingly simple but revealing task: *visual equation solving*. Given an image containing a system of equations where variables are depicted as object icons (e.g., 🍎🍎 + 🍌 = 10), the goal is to infer coefficients by counting icons and solve the equation accordingly. While this task appears tractable for models proficient in both visual and symbolic reasoning, our results show that even the strongest VLMs fail to solve such problems reliably. *Why do VLMs struggle with visual equation solving?* To answer this, we decompose the task into two core components: *symbolic equation solving* and *visual recognition*.

We begin by testing symbolic reasoning in isolation. When equations are presented in plain text in the image, VLMs solve them almost perfectly, confirming their mathematical reasoning and OCR capabilities. Next, we evaluate whether variable recognition is the bottleneck. Models are able to correctly identify object-based variables with high accuracy, suggesting recognition alone is not the issue. We then turn to coefficient estimation, counting the number of object instances. In hybrid settings where variables are icons and coefficients are numerals, or where both are visual, performance drops significantly. Direct evaluation of object counting further confirms that this is the key bottleneck: *VLMs often fail to infer quantities from repeated visual elements.*

Beyond counting, we observe that performance degrades further when multiple abilities, such as recognition and reasoning, must be composed. For instance, even when a model can recognize variables and solve symbolic equations separately, solving equations with icon-based variables and numeric coefficients proves difficult. This highlights compositional reasoning as another major challenge for current VLMs. Finally, we evaluate systems of equations with three variables. Even when

[1]We will release our data and code after the review process.

equations are presented symbolically, performance drops sharply, indicating that VLMs' mathematical reasoning is itself limited when faced with more complex problem structures.

Taken together, our findings reveal key limitations in current VLMs' ability to integrate perception and symbolic reasoning. In particular, visual counting and ability composition emerge as core bottlenecks, alongside limited generalization in symbolic math reasoning for complex tasks.

2 Preparation

We design a controlled evaluation setup to analyze VLMs' ability to perform visual equation solving. This section describes our data generation process and the experimental settings used for the following model evaluation.

2.1 Data

We construct synthetic visual math problems based on systems of linear equations, where variables are depicted as object icons and coefficients must be inferred from visual repetition. Each experiment is conducted on a set of 1,000 constructed examples and run once per model-setting configuration.

Equation Generation. We generate solvable systems of linear equations with unique integer solutions using matrix algebra, ensuring invertibility. To control visual complexity, coefficients are restricted to positive integers no greater than 10, limiting the number of repeated icons per image. All equations involve only addition, avoiding negative or fractional values. This setup ensures consistency and interpretability across all samples.

Image Construction. To visually represent equations, we map each variable to an icon selected from a curated set of 28 object types in the IconQA dataset (Lu et al., 2021), including items such as apples, bananas, flowers, and footballs. The coefficient of each variable is represented by repeating the corresponding icon the appropriate number of times. This creates visually grounded equations that require both recognition and symbolic reasoning. An example is shown in Fig. 1, and the full list of icons is provided in App. B.2.

2.2 Settings

Model List. We evaluate both proprietary and open-source VLMs. The former include GPT-4o (Hurst et al., 2024) and Gemini 2.0 Flash (Team et al., 2024), accessed via API. The latter consist of four models from the QwenVL-2.5 family (Bai et al., 2023), ranging from 3B to 72B parameters. To ensure fairness, all models are evaluated without batching, avoiding potential artifacts from cached context or batch-level optimizations. More details about the model can be found in App. B.3.

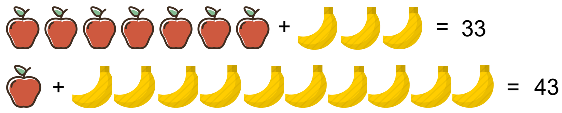


Figure 1: An example of our generated visual equations (i.e., systems of 2 linear equations with 2-variables).

Prompting Strategy. We apply two prompting strategies: direct zero-shot prompting (Direct) and two-step chain-of-thought (CoT) prompting. In the CoT setting, the model is first asked to extract the equation in free-form, then solve it in a second step. This setup encourages intermediate reasoning before committing to an answer. Both strategies are applied consistently across all models. Prompt templates and examples are provided in App. B.4.

Metrics. We evaluate accuracy by exact matching between the model-predicted variable values and the ground truth. We expect models to correctly associate each object type with its corresponding value and solve the equation.

3 Evaluation

We evaluate the mathematical reasoning capabilities of VLMs through the task of visual equation solving. Specifically, we investigate two research questions: (1) Can VLMs solve equations when they are visually grounded? (2) If not, what specific limitations hinder their performance?

3.1 Can VLMs Solve Equations?

We begin our evaluation on solving systems of linear equations in two formats: (1) a fully visual format, where both variables and coefficients are depicted visually (Fig. 1), and (2) a symbolic format, where equations are rendered as text within the image (Fig. 3). This comparison can isolate the impact of visual understanding on performance.

3.1.1 Visual Equation

Experiment Preparation. We use a default setting of two-variable linear equations with integer solutions. In each equation, variables are represented by object icons, and coefficients are conveyed by the number of repeated instances of each icon. This setup tests whether VLMs can integrate visual perception and symbolic reasoning.

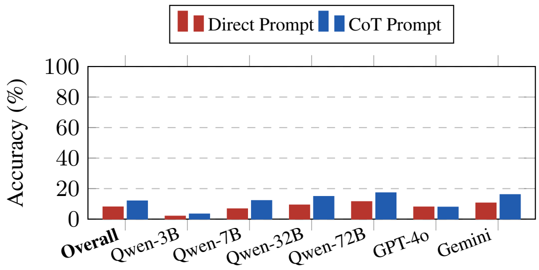


Figure 2: Performance of VLMs on visual equation solving. Results show that all models consistently fail to solve the equations correctly across both settings.

Results and Analysis. As shown in Fig. 2, all evaluated models, both proprietary and open-source, consistently fail to solve equations in visual form (overall accuracy $< 12\%$), despite their strong performance on other math and reasoning benchmarks. To rule out flaws in the evaluation setup, we include qualitative model outputs in App. C.1. These results raise a key question: Is the failure due to a lack of symbolic math reasoning, or a difficulty in interpreting equations visually?

3.1.2 Symbolic Equation

7 a + 3 b = 33 ; 1 a + 10 b = 43

Figure 3: An example of a system of linear equations represented in symbolic (textual) form.

Experiment Preparation. To isolate symbolic reasoning ability, we present the same equations in textual form within images (Fig. 3). If models succeed here, it would suggest that the core issue lies in interpreting the visual input, not solving the equations themselves.

Results and Analysis. Fig. 4 shows that all models, including the smallest Qwen-3B, achieve near-perfect accuracy on symbolic equations (accuracy $> 97\%$ with the CoT prompting). This confirms two things: (1) VLMs possess the required mathematical reasoning capabilities, and (2) they have strong OCR skills for extracting text from images. These findings indicate that the failure in the visual setting stems from difficulties in interpreting and grounding visual equations.

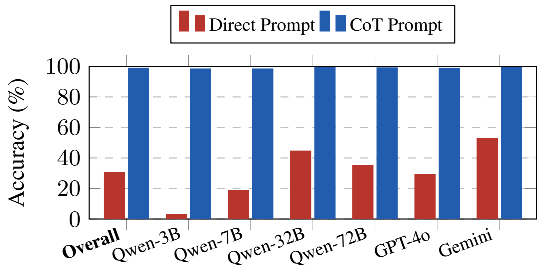


Figure 4: Performance of VLMs on symbolic equation solving. Results show that all models could solve the equations perfectly across settings.

3.2 Visual-Symbolic Gap Analysis

To understand the source of the performance gap between visual and symbolic settings, we decompose visual equation solving into two core sub-skills: (1) recognizing variables from icons, and (2) estimating coefficients by counting repeated visual instances. This allows us to evaluate whether recognition or counting is the main bottleneck, or whether it arises from composing the two abilities.

3.2.1 Coefficient Counting

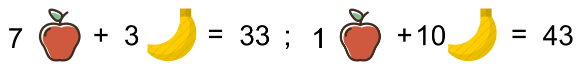


Figure 5: An example of our generated visual-symbolic equation, where the variable is denoted by icon but the coefficient is represented by symbolic number.

Experiment Preparation. We design a hybrid variant called visual-symbolic equations (Fig. 5), where variables are represented as icons, but coefficients are given as numeric text. This setting removes the need for counting while preserving the need for icon recognition and symbolic reasoning.

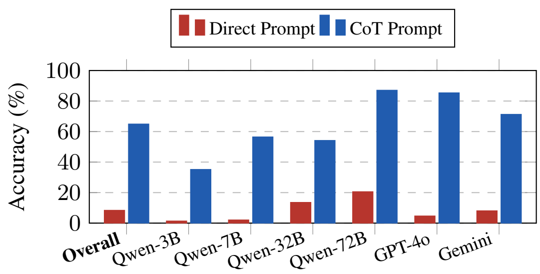


Figure 6: Performance of VLMs on visual-symbolic equation solving, where the coefficients are represented by symbolic numbers and variables are denoted by icons. Results show that all models could solve most systems of equations correctly.

Results and Analysis. As shown in Fig. 6, VLMs perform better in this setting than in the fully visual case (with overall accuracy as 64.45%), suggesting that coefficient counting is a major obstacle. To further confirm this, we directly evaluate models on isolated counting tasks (see App. C.2). These results clearly identify counting as a primary bottleneck in visual equation solving.

3.2.2 Variable Recognition

Experiment Preparation. To assess whether variable recognition contributes to the performance gap, we evaluate the ability to identify icon-based variables independently of counting. This task isolates visual recognition from symbolic reasoning.

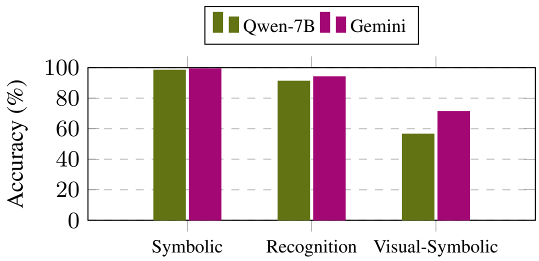


Figure 7: Accuracy on variable coefficient counting. Results show that both Qwen-7B and Gemini (under the CoT prompt) have difficulty to count the correct value of coefficients.

Results and Analysis. Fig. 7 shows that both Qwen-7B and Gemini achieve high accuracy in recognizing variables from icons (with accuracy above 90%), with performance comparable to symbolic settings. Details of prompt design are in App. B.4. This indicates that recognition itself is not a major limitation. Instead, the remaining gap between symbolic and visual-symbolic settings is likely due to task composition, i.e., the challenge of integrating recognition with downstream reasoning.

3.3 Three-Variable Equation

To assess the limitations of VLMs under increased mathematical complexity, we extend our evaluation to systems of three linear equations with *three variables*, which demand more advanced symbolic reasoning and variable tracking than the simpler two-variable case.

Experiment Preparation. We generate equations in the same formats as in the default setting: symbolic, visual-symbolic, and fully visual. This allows us to assess whether performance degradation stems from visual perception (i.e., recognition and counting) or from limitations in mathematical reasoning. We report the results under the CoT prompt as it achieves better performance.

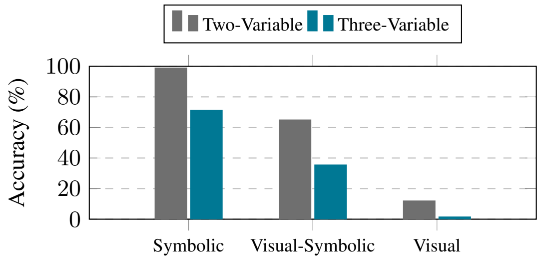


Figure 8: Overall accuracy across 6 models on solving equations with 2 and 3 variables. Results show that the bottleneck shift from vision side to the math reasoning.

Results and Analysis. As shown in Fig. 8, model performance drops significantly when moving from two-variable to three-variable systems (accuracy drops from 98% to 70% for the symbolic setting, and from 64% to 35% for the visual-symbolic setting). While the visual bottleneck remains largely unchanged, the additional complexity leads to a clear decline in symbolic reasoning. This indicates that, beyond perceptual limitations, current VLMs lack robust mathematical capabilities to solve more complex equation systems.

3.4 Takeaways.

Our experiments results show that VLMs perform well on symbolic equations but consistently fail on visual ones. The main bottleneck is visual counting, while variable recognition is largely accurate. However, composing recognition with reasoning introduces significant errors. As equation complexity increases, even symbolic reasoning begins to falter, revealing limits in the models' understanding.

4 Discussion

This paper investigates the reasoning limitations of VLMs through visual equation solving, a task that requires combining perception, counting, and symbolic computation. While VLMs perform well on symbolic equations and can reliably recognize visual variables, they fail when coefficients must be inferred from repeated visual instances. Our analysis identifies counting and ability composition as key bottlenecks, with performance degrading further as equation complexity increases.

These results highlight gaps in both visual grounding and symbolic reasoning. Addressing them may require new training objectives, compositional architectures, or integration with external tools. Our benchmark provides a diagnostic lens for understanding and improving VLMs on grounded, multi-step reasoning tasks.

Limitations

While our study provides insights into the mathematical reasoning capabilities of VLMs, it is subject to a few limitations. First, our evaluation focuses primarily on linear equations with integer solutions and addition-only operators. Although this setup allows controlled analysis, it does not capture the full spectrum of mathematical reasoning, such as non-linear or multi-operator problems. Second, while we isolate key sub-skills like counting and recognition, our diagnostic tasks are still synthetic and could not fully reflect real-world scenarios involving noisy or diverse visual contexts. Finally, we rely on prompting-based evaluation, which may under-represent the full potential of models fine-tuned for structured reasoning or equipped with external tools.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Jinze Bai, Rui Wang, Xiaoran Liu, Wuze Cong, Zicheng Wen, Yuying Cui, Shaohan Huang, Junjie Zhang, Xin Jiang, and Qun Liu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16678*.

Deepanway Ghosal, Navonil Majumder, Roy Lee, Rada Mihalcea, and Soujanya Poria. 2023. Language guided visual question answering: Elevate your multimodal language model using knowledge-enriched prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12096–12102, Singapore. Association for Computational Linguistics.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *arXiv preprint arXiv:1902.09506*. Published as a conference paper at CVPR 2019 (oral).

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024a. TopViewRS: Vision-language models as top-view spatial reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA. Association for Computational Linguistics.

Zhiyuan Li, Dongnan Liu, Chaoyi Zhang, Heng Wang, Tengfei Xue, and Weidong Cai. 2024b. Enhancing advanced visual reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1915–1929, Miami, Florida, USA. Association for Computational Linguistics.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-LLaVA: Bootstrapping mathematical reasoning for multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4663–4680, Miami, Florida, USA. Association for Computational Linguistics.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Zhuolin Yang, Wei Ping, Zihan Liu, Vijay Korthikanti, Weili Nie, De-An Huang, Linxi Fan, Zhiding Yu, Shiyi Lan, Bo Li, Mohammad Shoeybi, Ming-Yu Liu, Yuke Zhu, Bryan Catanzaro, Chaowei Xiao, and Anima Anandkumar. 2023. Re-ViLM: Retrieval-augmented visual language model for zero and few-shot image captioning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11844–11857, Singapore. Association for Computational Linguistics.

Yifan Zhang, Yuxuan Wang, Chunyuan Li, Yujia Chen, Jiacheng Liu, Pan Lu, Kai-Wei Chang, and Jianfeng Gao. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness in vision-language models. *arXiv preprint arXiv:2411.00836*.

A Related Work

Most existing benchmarks for evaluating VLMs treat perception and reasoning as separate capabilities, rather than testing them as a sequential, integrated process. Recognition-focused datasets such as VQA (Antol et al., 2015), GQA (Hudson and Manning, 2019), and CLEVR (Johnson et al., 2017) involve only minimal or trivial arithmetic, which current vision backbones can typically solve with ease. More recent efforts like MathVista (Lu et al., 2024) and DynaMath (Zhang et al., 2024) introduce a wider range of visual math problems, but they do not specifically evaluate whether models can solve algebraic equations where symbolic variables and coefficients are visually embedded. The ability to ground a visual system of equations and perform multi-step reasoning over visual cues remains largely untested.

B Details of Experiment Settings

B.1 Data License

All data used in this study is released under the CC BY 4.0 license. Each generated image is paired with a corresponding question that involves solving one or more equations, along with the ground-truth answers. Users are free to share, adapt, and build upon the dataset, provided appropriate credit is given.

B.2 Icon List

All the 28 icons that we use are listed below. For each icon, we use only one image to denote the object. Specifically, we select 28 icons labels randomly from the IconQA dataset, and for each label we randomly select one image icon. The icons are: *apple*, *palm_tree*, *strawberry*, *egg*, *clover*, *donut*, *mushroom*, *acorn*, *lemon*, *football*, *flower*, *sheep*, *panda*, *muffin*, *apricot*, *eggplant*, *broccoli*, *rabbit*, *banana*, *rubber_duck*, *horse*, *fish*, *tomato*, *candy*, *ice_cream_cone*, *cake*, *orange*, *carrot*.

B.3 Model Usage

We conduct inference using four NVIDIA H100 GPUs for each open-source VLM, including Qwen-3B, Qwen-7B, Qwen-32B, and Qwen-72B. All models are loaded using Hugging Face's Transformers library with automatic mixed-precision (torch.float16 or bfloat16) and memory-efficient device_map="auto" configurations. For each model, we adopt a consistent prompting strategy that combines images and text within structured chat templates. Inputs are tokenized and batched via model-specific processors. Inference is performed on individual image-equation instances using a maximum token length of 2048. We evaluate model outputs using an exact match criterion, comparing extracted variable assignments against ground-truth coefficients. To ensure fairness, we avoid prompt tuning and caching, and run each model independently on the same test set with uniform I/O and decoding procedures. Inference time ranges from 6 to 28 hours depending on model size, with Qwen-72B requiring the longest runtime.

B.4 Prompt

Direct Prompting. The direct prompt expects models to produce structured outputs in a single step. In our experiments, omitting object labels from the prompt led to poor generation quality, whereas including them significantly improved the reliability and evaluability of the outputs. The prompt template and an example are shown in Fig. 9.

Direct Prompt

You are given an equation image. Identify all icon types present in the image and determine their corresponding numerical values. Return **only** the icon type assignments in the format: `icon_type = number`. **For example:** `apple = 5, ice_cream_cone = 3`
Do not include any other text in your response. Only the following icon types are allowed: `apple, palm_tree, strawberry, egg, clover, donut, mushroom, acorn, lemon, football, flower, sheep, panda, muffin, apricot, eggplant, broccoli, rabbit, banana, rubber_duck, horse, fish, tomato, candy, ice_cream_cone, cake, orange, carrot`.

Figure 9: **Direct Prompting Template and Example.** The same prompt is used across all models for consistency.

Two-Step CoT Prompting. To encourage deeper reasoning while avoiding overly rigid output structures, we adopt a two-step chain-of-thought (CoT) prompting strategy. In the first turn, the model is prompted to freely analyze and solve the problem in its own words. In the second turn, we provide both the original prompt and the model's response, and ask it to extract the final answer. This separation between reasoning and answer extraction allows the model to engage in more flexible, interpretable analysis before committing to a structured output. The prompt used for the object-encoded benchmark is shown in Fig. 10.

Step 1: Analysis Prompt

Look at this equation image and identify all icon types and their corresponding values. Identify the objects, determine the mathematical operations, and solve the equation step-by-step.
Only use the following allowed objects: `apple, palm_tree, strawberry, egg, clover, donut, mushroom, acorn, lemon, football, flower, sheep, panda, muffin, apricot, eggplant, broccoli, rabbit, banana, rubber_duck, horse, fish, tomato, candy, ice_cream_cone, cake, orange, carrot.`

Step 2: Final Answer Prompt

Given the analysis: {Look at this equation image and identify all icon types and their corresponding values. Identify the objects, determine the mathematical operations, and solve the equation step-by-step...}, provide the final value of each identified object.
Respond only in the format: `object = value`.
For example: `flower = 5, carrot = 3`
Important: Do not include any other text. Only use allowed object names.

Figure 10: **CoT Prompting Strategy.** The left box initiates free-form reasoning, while the right box extracts the final answers based on the initial prompt and generated response.

Counting Prompting (CoT). An example input and prompt used for two-step prompting is shown in Fig. 11. This prompt is adapted from the CoT strategy and tailored for counting questions involving a single equation, rather than a full system of equations.

Step 1: Analysis Prompt

Look at this image and identify the count of each object. Provide your analysis step by step and ensure all details are clear. Only use the following allowed objects: `apple, palm_tree, strawberry, egg, clover, donut, mushroom, acorn, lemon, football, flower, sheep, panda, muffin, apricot, eggplant, broccoli, rabbit, banana, rubber_duck, horse, fish, tomato, candy, ice_cream_cone, cake, orange, carrot.`

Step 2: Final Answer Prompt

Now extract the final answer in the format: `object = number`. **For example:** `apple = 5, ice_cream_cone = 3`.
Do not include additional text. Only use allowed object names.

Figure 11: Two-step prompting strategy for solving visual object counting task.

Step 1: Analysis Prompt

Look at this image and identify the type of each object. Provide your analysis step by step and ensure all details are clear. Only use the following allowed objects: `apple, palm_tree, strawberry, egg, clover, donut, mushroom, acorn, lemon, football, flower, sheep, panda, muffin, apricot, eggplant, broccoli, rabbit, banana, rubber_duck, horse, fish, tomato, candy, ice_cream_cone, cake, orange, carrot.`

Step 2: Final Answer Prompt

Now extract the final answer in the format: `object = number`. **For example:** `apple,ice_cream_cone`.
Do not include additional text. Only use allowed object names.

Figure 12: Two-step prompting strategy for the object-type recognition task.

Recognition Prompting (CoT). An example input and prompt used for two-step prompting is shown in Fig. 11. This prompt is adapted from the CoT strategy and tailored for recognizing the object-type present in the image. We utilize the same single equation dataset as mentioned for the counting task.

B.5 Data Collection and Anonymization Procedures

To ensure ethical use and privacy compliance, we carefully selected data from publicly available, open-access visual datasets that contain no personally identifiable information or sensitive content. Specifically, we constructed our benchmark using synthetic images generated from programmatically defined math equations and object icons sourced from the IconQA dataset (Lu et al., 2021). These icons depict generic, non-human items such as apples, bananas, and lemons, eliminating the risk of capturing real-world identities or offensive imagery. Furthermore, we manually reviewed a sample of the generated images to confirm the absence of inappropriate or sensitive elements.

C Supplementary Results

C.1 Case Study

An example of an intermediate generation output is shown in Fig. 13. Upon closer inspection, we find that in most failure cases, the model incorrectly determines the coefficients during equation interpretation.

C.2 Can VLMs Count?

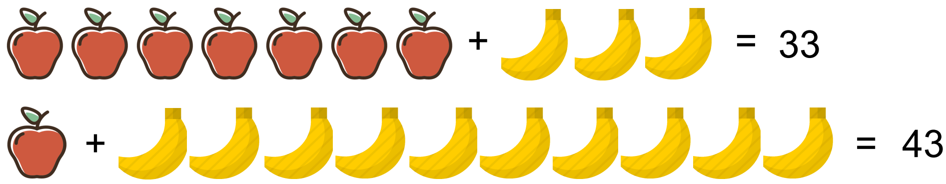
Fig. 14 shows the results of directly evaluating counting ability in the context of equation solving. In this setting, models are required to determine the coefficient of a variable by counting the number of repeated object icons. The corresponding prompt and template design are provided in App. B.4. As shown, all VLMs struggle significantly with this task under both direct and CoT prompting. Although CoT prompting provides a noticeable improvement across all models, the absolute performance remains far below acceptable levels, especially for the smaller open-source models. Notably, even advanced API-based models like GPT-4o and Gemini fail to reach consistent accuracy. This suggests that despite having strong recognition and reasoning abilities in isolation, VLMs are not yet capable of reliably counting visual instances, a key skill required for grounded symbolic reasoning. These results confirm that counting is the primary bottleneck limiting model performance on visual equation solving tasks.

D Potential Risk

Our study involves the generation of synthetic visual math equations using object icons, and evaluation is conducted using publicly available open-source models and commercially accessible API-based VLMs. As our work does not involve real-world data, human subjects, or sensitive content, we do not anticipate any ethical concerns or foreseeable risks associated with this research.

E Use of AI Assistants in Research

In our study, AI assistants were used sparingly and in accordance with ACL's Policy on AI Writing Assistance. We utilized ChatGPT and Grammarly for basic paraphrasing and grammar checks, respectively. These tools were applied minimally to ensure the authenticity of our work and to adhere strictly to the regulatory standards set by ACL. Our use of these AI tools was focused, responsible, and aimed at supplementing rather than replacing human input and expertise in our research process.



Gemini 2.0 Flash Response

Step 1: Object Identification
apple = x
banana = y

Step 2: Operation Analysis (Wrong Counting)

$$\left.\begin{aligned} 6x + 3y &= 33 \\ x + 9y &= 43 \end{aligned}\right\}$$

Wrong count! The original equations are
$7x + 3y = 33$
$1x + 10y = 43$

Step 3: Calculation
Substitute into second equation:

$$x = \frac{33 - 3y}{6}$$

$$\frac{33 - 3y}{6} + 9y = 43 \Rightarrow 33 - 3y + 54y = 258 \Rightarrow 51y = 225 \Rightarrow y = \frac{225}{51} = \frac{75}{17} \approx 4.41$$

Now, solve for x:

$$x = \frac{33 - 3 \cdot \frac{75}{17}}{6} = \frac{33 - \frac{225}{17}}{6} = \frac{\frac{561 - 225}{17}}{6} = \frac{336}{17 \cdot 6} = \frac{56}{17} \approx 3.29$$

Trial with integer values:

$$y = 5 \Rightarrow x = \frac{33 - 15}{6} = 3$$

Check:

$$6 \cdot 3 + 3 \cdot 5 = 33 \quad (\checkmark) \quad 3 + 9 \cdot 5 = 48 \quad (\text{X})$$

Alternative derivation by elimination:

$$6x + 54y = 258 \quad (\text{eq2} \times 6)(6x + 54y) - (6x + 3y) = 258 - 33 \Rightarrow 51y = 225 \Rightarrow y = \frac{75}{17}$$

$$x = 43 - 9 \cdot \frac{75}{17} = \frac{731 - 675}{17} = \frac{56}{17}$$

Step 4: Final Answer

$$\boxed{\text{apple} = \frac{56}{17} \quad \text{banana} = \frac{75}{17}}$$

Figure 13: Response of Gemini 2.0 Flash for the given query image (top) using the CoT prompting. Note the mistake made by the model in coefficient counting step (bottom). This cascades to the model eventually giving the wrong answer. The correct answer to this problem is apple (x) = 3, banana (y) = 4.

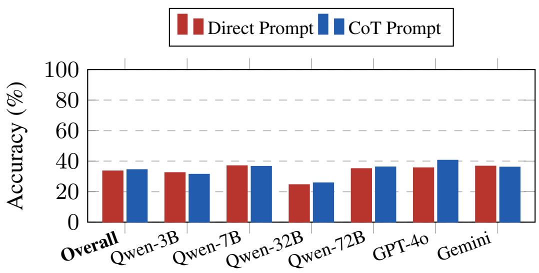


Figure 14: Performance of VLMs on variable coefficient counting. Results show that all models have difficulty to count the correct value of coefficients.