Self-Questioning Vision-Language Models: Reinforcement Learning for Compositional Visual Reasoning

Saraswathy Amjith
Massachusetts Institute of Technology
swathy@mit.edu

Code: github.com/saraswathyamjith/sq-vlms

Abstract

Vision-Language Models (VLMs) are AI systems that process both images and text, yet they often struggle with compositional visual reasoning questions that require chaining multiple steps together, such as identifying objects, counting them, and comparing the results. Existing approaches improve this reasoning by training models on human-written step-by-step explanations, but creating these annotations is expensive and difficult to scale. We propose a self-questioning framework that trains a VLM to break visual questions into smaller sub-questions and answer each one before producing a final response, using a reinforcement learning algorithm called Group Relative Policy Optimization (GRPO). The model is never shown examples of how to decompose questions, it discovers this behavior on its own, guided by a reward signal that scores whether the output is properly structured and whether the final answer is correct. We apply this framework to a 3-billion-parameter model, training on both synthetic scenes of geometric shapes (CLEVR) and real-world photographs (A-OKVQA). On A-OKVQA, both self-questioning and standard reinforcement learning substantially improve accuracy over the untrained model (52.2% and 51.6% vs. 46.8%). However, self-questioning introduces a cost on simple tasks, where the overhead of generating sub-questions can hurt rather than help. These results suggest that teaching AI systems to ask themselves intermediate questions is a promising strategy for complex visual reasoning, particularly when the difficulty of a question warrants explicit step-by-step decomposition.

1. Introduction

Vision-Language Models (VLMs) are AI systems that jointly process images and text, allowing a single model to look at a photograph and answer questions about what it sees. Recent advances in this area include models such as GPT-4V [12], LLaVA [10], and the Qwen-VL family [1] which can now caption images, answer open-ended questions about photographs, and follow complex multimodal instructions with impressive fluency [9]. These capabilities have allowed for deployment in real-world applications ranging from medical image interpretation to autonomous driving, because a model that can reliably understand images could automate tasks that currently require trained human experts.

Yet much of this success involves questions that can be answered through direct pattern recognition, identifying an object, reading a label, or describing a scene. A qualitatively harder class of problems, compositional visual reasoning, remains unsolved. Consider the question, "Are there more red cubes than blue spheres?" about a photograph of a cluttered tabletop. A correct response requires that the viewer first identify and count the red cubes, then identify and count the blue spheres, and finally compare the two quantities. Compositional visual reasoning is crucial to real-world applications: a medical imaging system must localize a lesion, assess its shape, and compare it to prior scans; an autonomous vehicle must identify a pedestrian, judge their trajectory, and decide whether to yield.

State-of-the-art VLMs fail on compositional questions because they produce answers in a single forward pass, that is, the model processes the image and question once and immediately outputs an answer, without any explicit intermediate reasoning.

Every existing approach to step-by-step visual reasoning depends on human-provided reasoning structure, limiting scalability. These strategies fall into three categories. First, chain-of-thought (CoT) prompting [7, 17] and its multimodal extensions [19, 20] supply the model with a handful of worked examples ("few-shot" examples) that demonstrate how to reason step by step, and then ask it to follow the same pattern on new questions. While effective, these methods are sensitive to which examples are

chosen: a poor selection can significantly degrade performance, and the model does not internalize a general reasoning strategy, it only imitates the specific patterns shown in the prompt. Second, supervised fine-tuning on human-written reasoning [2, 11] updates the model's weights by training it on datasets where humans have written out each reasoning step. This requires expensive, step-by-step annotations for every training example, a bottleneck that scales poorly to large datasets. Third, visual programming approaches [4, 16] translate each question into a short computer program that calls specialized vision tools (for example, an object detector or a counter) in sequence. These methods achieve strong compositional accuracy, but rely on hand-designed tool libraries that must be built in advance, making them lose effectiveness when faced with novel question types and generalization.

Reinforcement learning (RL) offers a fundamentally different training strategy that can improve reasoning without explicit human-created reasoning traces. Unlike supervised approaches, which teach a model by showing it the "right" answer to imitate, RL lets the model generate multiple candidate responses to the same question, scores each response with a numerical reward (for example, 1 for a correct final answer and -1 for an incorrect one), and then adjusts the model's internal parameters so that higher-scoring responses become more likely in the future while lower-scoring ones become less likely. In the text-only domain, DeepSeek-R1 [3] demonstrated that a specific RL algorithm called Group Relative Policy Optimization (GRPO) [15] can train large language models (LLMs) to generate intermediate reasoning steps without any human-annotated reasoning traces, simply by rewarding correct final answers produced in a structured format. Intuitively, GRPO works by sampling a group of candidate responses for each prompt, scoring them against a reward signal, and then updating the model to favor higher-scoring responses relative to the group—eliminating the need for a separate critic or value model.

We introduce a self-questioning framework in which a VLM is trained via GRPO to break visual questions into a sequence of sub-questions, answer each sub-question by attending to the image, and then synthesize a final response. Concretely, we require the model to produce output in a structured format, alternating `<sub_q>` (sub-question) and `<sub_a>` (sub-answer) tags, but we never show it example decompositions during training. Instead, we define a reward function (the numerical signal that guides RL training) that grants a score of 1.0 if and only if the model (a) produces output in the prescribed sub-question format and (b) arrives at the correct final answer, or a score of -1.0 otherwise. The decomposition behavior must therefore emerge entirely from the training process itself, without any human demonstrations of how to decompose questions.

We evaluate whether self-questioning emerges and transfers by training on two benchmarks and testing across domains. Our base model is Qwen2.5-VL-3B-Instruct [1], a 3-billion-parameter VLM that accepts both image and text inputs. We chose this model because its compact size makes RL training feasible on limited hardware, while still being representative of the VLM family. We train and evaluate on two benchmarks: CLEVR [6], a synthetic dataset of geometric scenes, and A-OKVQA [14], a real-world visual question answering dataset (details in Section 3.6).

Our experiments reveal that RL training is the primary driver of improvement: both self-questioning and direct RL raise A-OKVQA accuracy from 46.8% to over 51%, with self-questioning providing only a modest additional gain (+0.6pp). A model trained with self-questioning on synthetic images does show some transfer to real-world photographs (+2.6pp over baseline), but structured decomposition also introduces a cost on simpler tasks, where the overhead of generating sub-questions can hurt rather than help. These results suggest that the value of self-questioning is contingent on question complexity, and that RL training itself, rather than the specific reasoning format, accounts for most of the observed gains.

Contributions. In summary, this work makes the following contributions:

- We propose a *self-questioning* framework that trains a VLM to decompose compositional visual questions into sub-question–answer pairs using reinforcement learning, without any reasoning demonstrations.
- We conduct controlled experiments comparing self-questioning against direct RL on both synthetic (CLEVR) and real-world (A-OKVQA) benchmarks, identifying both the benefits and disadvantages of structured decomposition.

2. Related Work

Our work sits at the intersection of vision-language modeling, compositional visual reasoning, step-by-step reasoning, and reinforcement learning for language models. Modern VLMs have advanced from image captioning to open-ended visual question answering, yet chaining multiple reasoning steps over an image remains a challenge. Step-by-step reasoning methods address this by introducing intermediate computations, but existing approaches depend on human-created demonstrations. Reinforcement learning offers an alternative, rewarding correct answers and allowing the model to discover its own reasoning structure.

Vision-Language Models Modern VLMs integrate visual encoders with large language models to process interleaved image and text inputs. Early models such as LLaVA [10] demonstrated that training on image–text pairs yields strong performance on visual question answering.

The Qwen-VL family [1] scaled this approach and GPT-4V [12] showed that sufficiently large multimodal models can exhibit compositional reasoning without any task-specific training. However, this compositional ability degrades significantly in smaller models, which tend to answer questions through pattern matching rather than multi-step reasoning. Our work uses Qwen2.5-VL-3B-Instruct, a compact 3-billion-parameter VLM, to study whether reinforcement learning can compensate for this limitation, whether a smaller model can be trained to reason through problems that it would otherwise answer incorrectly.

Step-by-Step Reasoning in Vision-Language Models Several lines of work have attempted to improve VLM reasoning by introducing intermediate steps between question and answer. Chain-of-thought (CoT) prompting [17] showed that language models produce more accurate answers when prompted to "think step by step," and zero-shot variants [7] demonstrated that this works even without worked examples. Multimodal extensions include Multimodal-CoT [19], which fine-tunes models on paired rationales and answers, and DDCoT [20], which uses prompting to separate the reasoning phase from the answering phase. A related line of work, self-taught reasoning (STaR) [18], trains models to generate and filter their own reasoning traces through an iterative process, reducing dependence on human annotations but still requiring an initial set of correct rationales. All of these approaches share a common limitation, they depend on either human-written reasoning demonstrations or carefully selected examples to guide the model's step-by-step behavior. Our framework removes this dependency entirely, the model discovers how to decompose questions through reinforcement learning, with no reasoning demonstrations provided at any stage of training.

Reinforcement Learning for Reasoning Reinforcement learning offers an alternative to supervised reasoning demonstrations: rather than showing a model how to reason, RL rewards it for arriving at correct answers and lets the reasoning strategy emerge from training. DeepSeek-R1 [3] demonstrated this in the text-only setting, showing that large language models develop chain-of-thought reasoning when trained with Group Relative Policy Optimization (GRPO) [15] and rewarded only for correct final answers in a structured format. GRPO works by sampling a group of candidate responses for each prompt, scoring them, and updating the model to favor higher-scoring responses relative to the group, eliminating the need for a separate critic model that estimates response quality. To our knowledge, our work is the first to apply GRPO to a vision-language model for compositional visual reasoning.

3. Method

This section describes our self-questioning framework for training VLMs to decompose visual questions into sub-questions using reinforcement learning. Figure 1 provides an overview of the RL setup.

3.1. Problem Formulation

Given an image I and a question q, a VLM generates a textual response $y = f_\theta(I, q)$. In standard VQA, the model produces a direct answer in one step. We instead train the model to generate a structured response consisting of sub-question–answer pairs followed by a final answer:

$$y = [\mathrm{SQ}_1, \mathrm{SA}_1, \ldots, \mathrm{SQ}_k, \mathrm{SA}_k, \mathrm{FA}] \tag{1}$$

where SQ_i and SA_i denote the i-th sub-question and its answer, k is the number of sub-questions (determined by the model, not by us), and FA is the final answer.

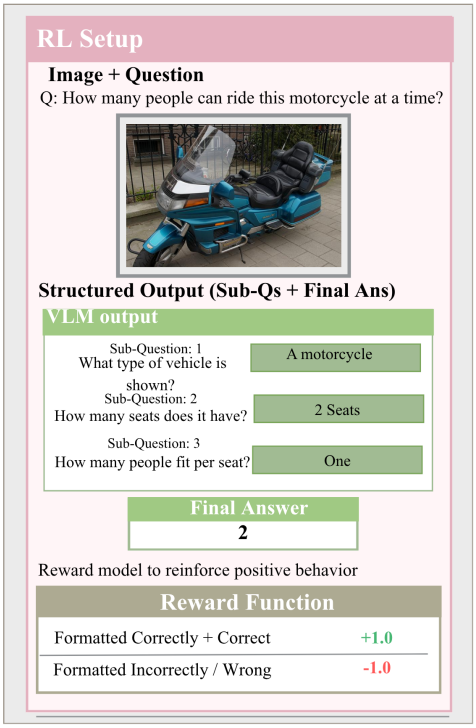

Figure 1. Overview of the self-questioning framework. Given an image and question, the VLM generates sub-questions, answers each by attending to the image, and produces a final answer. The reward function requires both correct format and correct answer; GRPO uses group-normalized advantages to update the policy.

3.2. Self-Questioning Prompt

We use a system prompt that instructs the model to generate sub-questions before answering.

```
You are given an image and a question.
Before answering, generate a series of
sub-questions to help you analyze the image
carefully.  For each sub-question, look at
the image and provide an answer.  Then give
your final answer.

Format:
Sub-question 1:  [question]
Answer 1:  [answer]
Sub-question 2:  [question]
Answer 2:  [answer]
...
Final Answer:  [answer]
```

We deliberately omit examples from the prompt because including them would bias the model toward imitating specific decomposition patterns rather than discovering its own.

3.3. Reward Function

Our reward function enforces format compliance and answer correctness. Given a model completion y and ground-truth answer a^*:

$$R(y, a^*) = \begin{cases} 1.0 & \text{if Format}(y) \wedge \text{Correct}(y, a^*) \\ -1.0 & \text{otherwise} \end{cases} \tag{2}$$

where Format(y) checks for at least one "Sub-question / Answer" pair, |FA| is the word count of the final answer, and Correct(y, a^*) uses containment matching, the normalized ground truth must appear as a substring of the normalized prediction (case-insensitive).

3.4. Training with GRPO

We use Group Relative Policy Optimization (GRPO) [15] as our RL algorithm. We chose GRPO over the more common Proximal Policy Optimization (PPO) [13] because GRPO eliminates the need for a separate value network (a second neural network that estimates how good each state is), reducing memory requirements by roughly half, a critical advantage when training on limited GPU hardware. Instead of learning a value baseline, GRPO normalizes rewards within groups of completions generated for the same prompt.

Concretely, for each training prompt (I, q), GRPO samples G completions $\{y_1, \ldots, y_G\}$ from the current model and computes group-normalized advantages:

$$\hat{A}_i = \frac{R(y_i, a^*) - \mu_g}{\sigma_g} \tag{3}$$

where μ_g and σ_g are the mean and standard deviation of rewards within the group. The policy is then updated using a clipped surrogate objective with a KL (Kullback-Leibler) divergence penalty against the reference policy π_{ref}:

$$\mathcal{L}(\theta) = -\mathbb{E}\Big[\min\big(r_t \hat{A}_t,\ \text{clip}(r_t, 1{-}\epsilon, 1{+}\epsilon)\hat{A}_t\big) - \beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big] \tag{4}$$

where $r_t = \pi_\theta(y_t)/\pi_{\text{ref}}(y_t)$ is the importance sampling ratio. The KL divergence measures how far the updated policy has drifted from the reference policy. The β-weighted penalty prevents the model from changing too heavily during training, without it, the model might improve at answering questions in the sub-question format but lose its underlying ability to understand images or produce coherent language, a phenomenon known as catastrophic forgetting.

3.5. Baseline: Direct RLVR

To isolate the specific effect of self-questioning from the general effect of RL training, we train a baseline model using the same GRPO procedure but with a simplified prompt that requests only a direct answer and a reward function that checks answer correctness and conciseness without any format requirement. This baseline is essential because it represents standard reinforcement learning from verifiable rewards (RLVR) applied to visual question answering. By using identical hyperparameters and training data for both models, any accuracy difference between them can be attributed specifically to the self-questioning format rather than to RL training in general.

3.6. Implementation Details

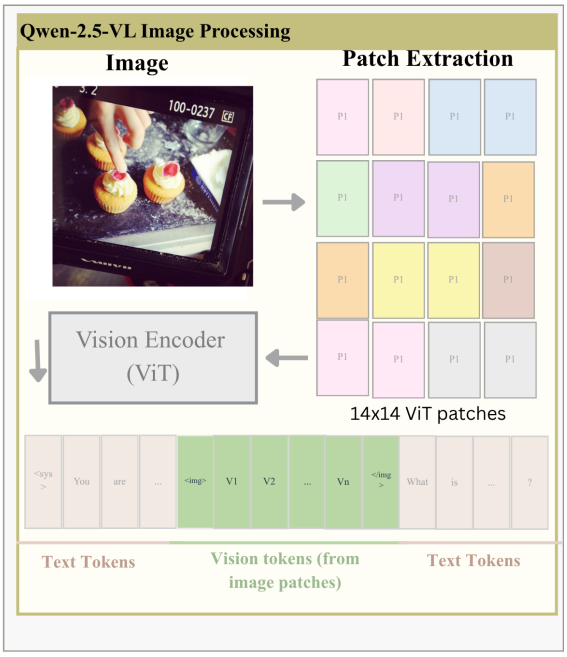

Figure 2. Overview of image processing in Qwen2.5-VL. The input image is divided into 14×14 patches, which are encoded by a Vision Transformer (ViT) into a sequence of vision tokens. These vision tokens are interleaved with text tokens in the model's input sequence, delimited by special <img> tokens.

Base model. We use Qwen2.5-VL-3B-Instruct [1], a 3-billion-parameter VLM. Its compact size makes RL training feasible on academic-scale hardware while still being large enough to exhibit meaningful visual reasoning capabilities.

Datasets. We train on two datasets: (1) a 10,000-example subset of CLEVR [6], a synthetic visual reasoning benchmark with compositional questions about computer-generated scenes of geometric objects; and (2) a 10,000-example subset of A-OKVQA [14], a challenging real-world visual question answering (VQA) dataset requiring outside knowledge and complex reasoning about natural photographs. CLEVR's simpler questions make it an ideal starting point for studying emergent visual reasoning, while A-OKVQA tests whether the framework generalizes to the greater complexity of natural images, which demand external knowledge beyond what is visually present. We selected CLEVR in part because of efficiency, its synthetic, simple scenes allow rapid training. We selected A-OKVQA because of its universality, its diverse real-world photographs and open-ended questions test whether models can generalize across the broad range of visual and knowledge domains that practical deployment demands.

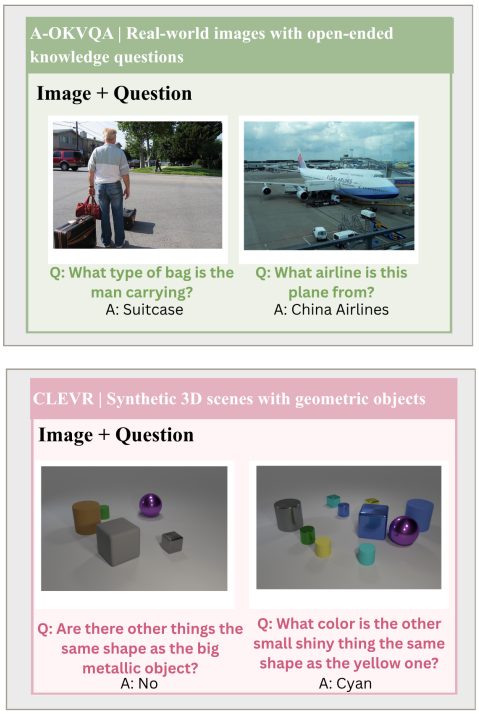

Figure 3. Examples from the two evaluation benchmarks.
Top: A-OKVQA, which pairs real-world images with open-ended questions requiring external knowledge.
Bottom: CLEVR, which poses compositional reasoning questions over synthetic 3D scenes with geometric objects.

Parameter-efficient fine-tuning. Rather than updating all 3 billion parameters during training, we apply Low-Rank Adaptation (LoRA) [5]. In standard fine-tuning, every weight in the model is adjusted, which requires storing a full copy of all parameter gradients in memory. LoRA instead freezes the original weights and learns a small pair of matrices for each targeted layer that together approximate the weight change needed for the new task. Because these matrix pairs are much smaller than the original weight matrices, LoRA updates less than 1% of the model's parameters while achieving performance comparable to full fine-tuning [5]. We use rank $r = 16$, $\alpha = 32$, and dropout 0.05, targeting the query and value projection matrices (`q_proj`, `v_proj`). This fits within the memory capacity of our GPUs (48 GB each), which full fine-tuning would exceed.

GRPO configuration. We train with batch size 1, gradient accumulation over 4 steps (effective batch size 4), learning rate 1×10^{-5}, maximum completion length of 256 tokens, and sampling temperature 0.7. We use G=8 generations per prompt for both datasets, as larger groups increase the likelihood of reward variance within a group, providing denser gradient signal for learning. Generation is accelerated using vLLM [8] with 70% GPU memory utilization on a dedicated GPU.

Hardware. Training runs on 2× NVIDIA L40S GPUs (48 GB each): one for model training and one for vLLM-based generation. We dedicate a separate GPU to generation because GRPO requires sampling multiple completions per prompt, which is memory-intensive. Checkpoints are saved every 500 steps.

4. Results

We designed experiments to answer three questions: (1) Does Reinforcement-Learning training improve Visual Question Answering accuracy? (2) Does self-questioning provide additional benefit beyond standard Reinforcement-Learning? (3) Does the learned reasoning behavior transfer across visual domains?

4.1. Training Setup

We evaluate five models (Table 3) to enable comparisons:

- **SQ+GRPO (CLEVR)**: self-questioning prompt, trained on CLEVR with G=8 for ∼7,000 steps.
- **Direct+GRPO (CLEVR)**: direct-answer prompt, trained on CLEVR with G=8 for ∼7,000 steps.
- **SQ+GRPO (A-OKVQA)**: self-questioning prompt, trained on A-OKVQA with G=8 for ∼7,000 steps.
- **Direct+GRPO (A-OKVQA)**: direct-answer prompt, trained on A-OKVQA with G=8 for ∼7,000 steps.

- **Base**: the unmodified Qwen2.5-VL-3B-Instruct model, with no RL training applied.

On each dataset, the SQ+GRPO and Direct+GRPO models use identical hyperparameters (Table 3) and training data, differing only in prompt format and reward function, ensuring any performance difference between the two models can be attributed to the self-questioning structure itself, rather than to differences in training data, compute, or optimization.

4.2. Evaluations

We evaluate on 500 examples from the validation set of each benchmark. We report:

- **Overall accuracy**: the fraction of correctly answered examples out of 500.
- **Prompt ablation**: each trained model is evaluated with both its trained-on prompt format and the alternative format (e.g., an SQ-trained model tested with a direct prompt, and vice versa). This distinguishes whether accuracy gains come from changes in the model's internal knowledge or simply from the prompt format used at test time.

We organize our results around three questions: does RL training improve accuracy, does self-questioning add benefit beyond standard RL, and does the learned reasoning transfer across domains?

4.3. A-OKVQA: Main Comparison

Table 1. A-OKVQA validation accuracy (500 examples). SQ+GRPO and Direct+GRPO are both trained on A-OKVQA for ∼7,000 matched steps. Both RL methods improve over the base model, with SQ providing a marginal advantage.

Model	Train Data	Eval Prompt	Accuracy
Base	—	direct	46.8
Direct+GRPO	A-OKVQA	direct	51.6
SQ+GRPO	A-OKVQA	sq	**52.2**
SQ+GRPO	A-OKVQA	direct	47.8
Direct+GRPO	A-OKVQA	direct	47.0
SQ+GRPO	CLEVR	sq	**49.4**
SQ+GRPO	CLEVR	direct	47.4

Table 1 compares all models on 500 A-OKVQA validation examples. SQ+GRPO and Direct+GRPO are both trained on A-OKVQA for ∼7,000 steps with identical hyperparameters, differing only in prompt format and reward function.

Table 2. CLEVR validation accuracy (500 examples). All models near ceiling on this benchmark. Self-questioning introduces a format tax: the A-OKVQA-trained SQ model drops to 81.0% with the SQ prompt but recovers to 97.6% with a direct prompt, confirming the model weights are intact.

Model	Train Data	Eval Prompt	Accuracy
Base	—	direct	97.4
Direct+GRPO	CLEVR	direct	98.4
SQ+GRPO	CLEVR	direct	**98.6**
SQ+GRPO	CLEVR	sq	97.2
Direct+GRPO	A-OKVQA	direct	**98.6**
SQ+GRPO	A-OKVQA	direct	97.6
SQ+GRPO	A-OKVQA	sq	81.0

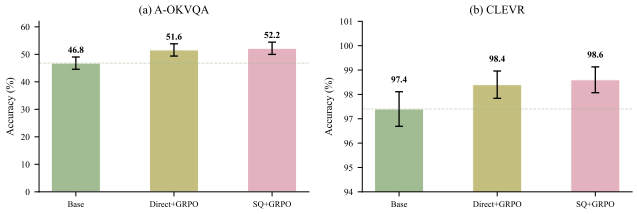

Figure 4. Accuracy comparison across A-OKVQA and CLEVR. Both RL methods improve over the base model on A-OKVQA, but the SQ format introduces a format tax on CLEVR that disappears with direct prompting—confirming the tax is a property of the prompt, not the model weights.

RL training improves accuracy. Both GRPO-trained models substantially outperform the base model (Figure 4): SQ+GRPO achieves **52.2%** (+5.4pp) and Direct+GRPO achieves **51.6%** (+4.8pp), compared to the base model's 46.8%. This improvement occurs because the RL reward signal teaches the model to calibrate its outputs toward correct answers: during training, the model generates multiple candidate responses, and those that produce correct short answers are reinforced while incorrect responses are suppressed. Over thousands of training steps, this shapes the model into a more reliable question-answerer.

Self-questioning provides marginal benefit. At matched training steps, SQ+GRPO (52.2%) slightly outperforms Direct+GRPO (51.6%), a difference of only 0.6 percentage points. This suggests that the accuracy gain comes primarily from the RL training signal rather than from the structured sub-question decomposition. One likely explanation is that A-OKVQA questions, while challenging, often do not require deep multi-step decomposition: many can be answered through strong visual recognition and world knowledge, which RL training improves regardless of output format. The training reward curves (Figure 5) illustrate how

on CLEVR, the Direct+GRPO model starts with a reward near 0.97, leaving almost no room for self-questioning to provide additional benefit. On A-OKVQA, both methods start lower and converge to similar reward levels by 7,000 steps,

Prompt ablation. When the A-OKVQA-trained SQ model is evaluated with a direct prompt instead of the SQ prompt, accuracy drops to 47.8%, nearly back to the base model level. This means the model's accuracy improvement is tied to the sub-question format: the model has learned reasoning strategies that are achieved through the sub-question structure, and removing that structure removes the benefit.

Cross-dataset transfer. The CLEVR-trained SQ model achieves 49.4% on A-OKVQA with the SQ prompt (+2.6pp over base), despite never seeing real-world images during training. CLEVR contains only synthetic scenes of geometric shapes, yet the self-questioning strategy learned on these simple images provides benefits on complex natural photographs. The result suggests that what transfers is the decomposition itself, rather than any domain-specific visual knowledge.

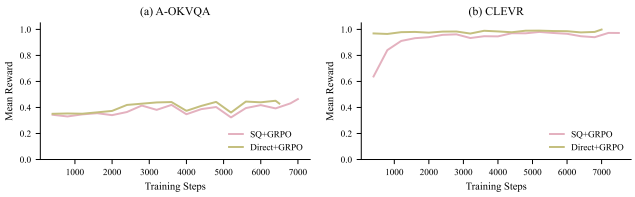

Figure 5. Training reward curves (G=8, 200-step rolling average). **(a)** On A-OKVQA, both methods gradually improve, with SQ+GRPO and Direct+GRPO reaching similar reward levels by 7,000 steps. **(b)** On CLEVR, Direct+GRPO starts near 0.97 while SQ+GRPO climbs from 0.64 as the model learns the structured format.

Capability loss on simpler tasks. Table 2 presents CLEVR validation results. Self-questioning introduces what we call a format tax: an accuracy cost from the overhead of generating sub-questions on problems simple enough to answer directly. The A-OKVQA-trained SQ model achieves only **81.0%** on CLEVR with the SQ prompt, a 16.4pp drop from the base model's 97.4%. However, when the same model is evaluated with a direct prompt, accuracy recovers to 97.6%, confirming that the model's visual capabilities are intact, the loss comes entirely from forcing decomposition on questions like "How many red cubes are there?" that the model can already answer in a single step. The CLEVR-trained SQ model pays a much smaller format tax: 97.2% with the SQ prompt vs. 98.6% with a direct prompt (only 1.4pp). Training on in-domain data teaches the model to generate minimal sub-questions that add little overhead, suggesting the tax is reducible with in-domain training.

4.4. Qualitative Analysis

To understand what the model actually learns to do, we examine individual model outputs. The SQ-trained model generates sub-questions for every response (100% format compliance), confirming that the format constraint is fully learned. On A-OKVQA, the sub-questions sometimes identify relevant aspects of the scene. For example, given "What are the people looking over?" (Figure **??**):

```
Sub-question 1: What are the people
in the image doing?
Answer 1: They are skiing down a
mountain.
Sub-question 2: Where are they
located?
Answer 2: They are on the slope.
Final Answer: Mountain peak.
```

Here, the sub-questions decompose into intermediate observations (activity, location) that support the final answer. However, on CLEVR the decompositions often restate the original question nearly verbatim. This likely reflects the simplicity of CLEVR questions: the base model can already answer most of them correctly in a single step, so there is no reasoning gap for sub-questions to fill.

5. Discussion

5.1. When Does Self-Questioning Help?

Our results suggest that self-questioning is most valuable when the question is complex enough that a single-step answer is unreliable, and the model is evaluated with the sub-question prompt it was trained on. When either condition is absent, simple questions (CLEVR) or a mismatched prompt (SQ model with direct prompt), the benefit disappears. Future work may benefit from exploring adaptive prompting strategies that apply self-questioning selectively based on estimated question difficulty, decomposing only when the overhead is likely to pay off.

Comparison with related approaches. To our knowledge, no prior work has applied GRPO to a vision-language model at the 3B-parameter scale. Our +5.4pp gain on A-OKVQA is achieved without human-authored reasoning traces, unlike supervised methods such as Multimodal-CoT [19] that require paired rationale annotations. DeepSeek-R1 [3] showed GRPO elicits chain-of-thought in text-only LLMs, our results partially replicate this in the multimodal setting. Visual programming methods [16] achieve strong compositional accuracy but depend

on hand-designed tool libraries, whereas our framework requires no external tools and transfers across domains.

5.2. Failure Cases

We identify two recurring failure modes in SQ+GRPO outputs.

Superficial sub-questions. The model sometimes generates sub-questions that paraphrase the original question rather than decomposing it. For example, given "What sport is being played?", the model may ask "What activity are the people doing?" and "What game is this?", i.e. restatements that do not extract new visual information. The sub-questions add token increase without contributing to accuracy.

Format-induced errors on simple questions. On CLEVR, the SQ format occasionally degrades accuracy on questions that the base model answers correctly in a single step. For example, a simple counting question ("How many red cubes are there?") may be answered correctly without sub-questions, but the SQ model introduces an unnecessary decomposition that leads to a miscounted intermediate answer, propagating the error to the final response. This failure mode explains the accuracy loss observed in Table 2.

5.3. Limitations

Model scale. We train a 3B-parameter model with LoRA. Larger models or full fine-tuning may show different results. Self-questioning could provide greater benefit for small models with weaker base reasoning, because those models have more room for improvement through structured decomposition. Conversely, very large models may already reason compositionally without explicit structure. Testing across model sizes (e.g., 7B, 14B, 72B variants in the Qwen-VL family) with full fine-tuning would reveal where self-questioning provides the greatest improvement.

Evaluation scope. We evaluate on two VQA benchmarks. Tasks requiring deeper multi-step reasoning may show larger benefits from structured decomposition, because the gap between what the model can answer in one step versus multiple steps is wider on harder tasks.

5.4. Ethical and Societal Considerations

Bias amplification. Our framework trains on existing VQA datasets. CLEVR is synthetically generated images of geometric objects and unlikely to carry demographic biases, but A-OKVQA draws on real-world photographs and crowdsourced annotations, which may contain biases in the types of questions asked, the images selected, or the answer distributions. However, the dataset creators report efforts to minimize such biases. RL training could amplify potential biases, because the reward function reinforces whatever answer patterns lead to high reward, regardless of whether those patterns reflect societal stereotypes. For example, if the training data disproportionately associates certain activities with specific demographics, the model may learn to reproduce these associations. Mitigation strategies include auditing training data for demographic biases before training, evaluating model outputs across demographic subgroups, and incorporating fairness constraints into the reward function.

Misuse potential. Any model capable of visual question answering could in principle be applied to surveillance or privacy-invasive tasks. Self-questioning does not fundamentally change this threat, but the structured sub-question output makes the model's information extraction more interpretable, and therefore easier to audit. We believe deployment of such models should include access controls and use-case restrictions regardless of whether self-questioning is used.

Transparency and reproducibility. We use a pre-trained open-weight model (Qwen2.5-VL-3B-Instruct) and publicly available datasets. All hyperparameters are reported in Table 3, and our code is available to support reproducibility.

6. Conclusion

We asked whether a VLM could learn to decompose compositional visual questions into sub-questions without any human demonstrations of step-by-step reasoning. Our self-questioning framework shows that the answer is yes, but with important caveats. RL training is the primary driver of accuracy improvement, raising A-OKVQA accuracy from 46.8% to over 51% regardless of whether self-questioning is used. The sub-question format adds only a marginal benefit (+0.6pp), and introduces a format tax on simpler tasks where decomposition creates overhead without aiding reasoning. These findings point toward a practical direction: adaptive systems that apply self-questioning selectively, decomposing only when question complexity warrants the overhead. More broadly, our results suggest during RL-based reasoning training for vision-language models, the format and structure of reasoning, not just the reward signal, will be an important design choice, one whose costs and benefits depend on task difficulty, model scale, and domain.

GenAI Disclosure. A generative AI tool (Claude, by Anthropic) was used only to assist with finding related articles and sources.

References

[1] Shuai Bai et al. Qwen2.5-vl technical report, 2025.

[2] Hanqi Chen et al. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *ACL*, 2024.

[3] Daya Guo et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025.

[4] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[6] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[7] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

[8] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023.

[9] Bohao Li et al. SEED-Bench: Benchmarking multimodal LLMs with generative comprehension, 2023.

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[11] Pan Lu et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[12] OpenAI. GPT-4V(ision) system card, 2023. Technical report.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[14] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *ECCV*, 2022.

[15] Zhihong Shao et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024.

[16] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *ICCV*, 2023.

[17] Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

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

[19] Zhuosheng Zhang et al. Multimodal chain-of-thought reasoning in language models, 2023.

[20] Ge Zheng et al. DDCoT: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In *NeurIPS*, 2023.

A. Training Hyperparameters

Table 3. GRPO training hyperparameters, matched across all models and datasets.

Hyperparameter	Value
Base model	Qwen2.5-VL-3B-Instruct
LoRA rank / alpha	16 / 32
LoRA targets	`q_proj`, `v_proj`
Learning rate	1×10^{-5}
Batch size (effective)	4
Generations per prompt (G)	8
Max completion length	256 tokens
Sampling temperature	0.7
Training steps	$\sim$7,000
Training examples	10,000
Optimizer	AdamW
Precision	bfloat16