# RISE: RELIABLE IMPROVEMENT IN SELF-EVOLVING VISION-LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision-language models (VLMs) have achieved strong multimodal reasoning capabilities, but further improving them still relies heavily on large-scale human-constructed supervision for post-training. Such supervision is costly to obtain, especially for reasoning-intensive multimodal tasks where questions, answers, and feedback signals must be carefully designed. This motivates self-evolving learning, where a model improves itself through a dual-role closed loop: a questioner autonomously poses questions and a solver learns to solve them. However, we observe that current VLM self-evolving methods still face three major challenges: coarse-grained role alternation delays the interaction between question generation and solver adaptation; generated questions can progressively degrade in quality; and question types may collapse toward a narrow distribution. These issues limit the efficiency and reliability of self-evolution. Thus, we propose **RISE**, a reliable self-evolving framework for vision-language models. RISE is built on three complementary designs: fine-grained role alternation, which shortens the feedback loop between the questioner and the solver to improve efficiency; a quality supervisor, which improves question validity and pseudo-label reliability; and skill-aware dynamic balancing, which mitigates mode collapse and maintains broad skill coverage during evolution. Together, these components enable more reliable and effective self-evolution from unlabeled images. Experiments on two VLM backbones across seven benchmarks show that RISE consistently improves the base models, yielding broad and sustained gains.

## 1 INTRODUCTION

Vision-language models (VLMs) have recently achieved substantial progress in multimodal understanding and reasoning (Li et al., 2024a; Bai et al., 2025a; Zhu et al., 2025; An et al., 2025). However, such progress has relied heavily on large-scale, high-quality human-annotated data. As increasingly powerful models (Achiam et al., 2023; Team et al., 2023; Guo et al., 2025; Liu et al., 2024) consume much of the high-quality training resources available on the Internet, further scaling model capability by relying on externally curated supervision is becoming increasingly costly and less scalable (Tong et al., 2024; Yin et al., 2024; Li et al., 2023; Yang et al., 2026a). This challenge is even more pronounced for VLMs: while large amounts of unlabeled images are readily available, high-quality visual reasoning supervision remains scarce, especially data that jointly couple images, questions, and answers in a reasoning-intensive manner (Villalobos et al., 2024; Li et al., 2024c; Zhang et al., 2024a). This raises an increasingly important question: can VLMs rely more on self-generated learning signals from unlabeled images to achieve sustained capability improvement (Deng et al., 2025a)?

In the large language model (LLM) domain, self-evolving frameworks such as R-Zero (Huang et al., 2025a) and Absolute Zero (Zhao et al., 2025) offer a promising route toward this goal (Xia et al., 2025; Liu et al., 2025; Yu et al., 2025). These methods cast learning as a closed-loop interaction between two complementary roles: one role proposes tasks near the current capability boundary, while the other attempts to solve them, so that the model can autonomously construct training signals and improve itself without relying on human annotations (Tao et al., 2024). Inspired by this paradigm, recent studies have begun to transfer this self-evolving paradigm to the vision-language setting. For example, works such as VisPlay (He et al., 2025) and MM-Zero (Li et al., 2026) extend this idea to visual inputs, enabling VLMs to improve themselves by learning from the self-generated data. This



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

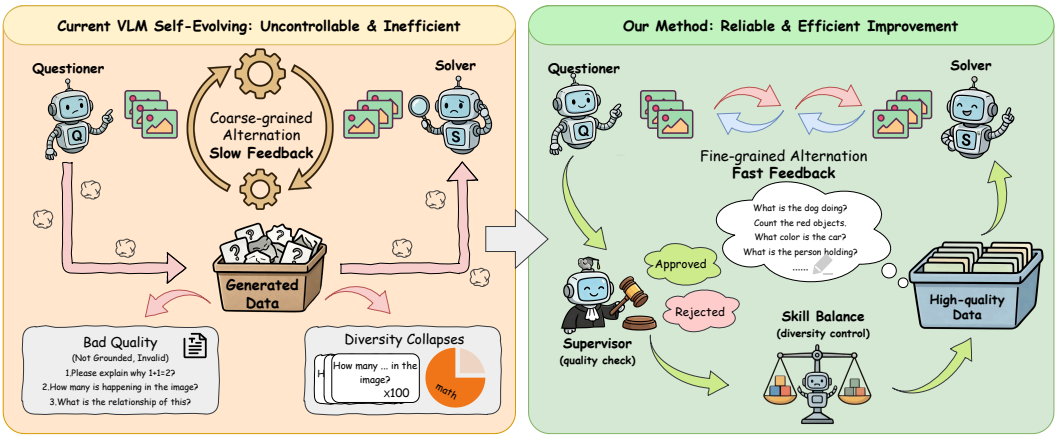


Figure 1: Comparison between current VLM self-evolving frameworks and our RISE framework.

migration from LLM self-evolution to VLM self-evolution provides a natural and appealing direction for unlabeled visual reasoning learning.

However, directly transferring such dual-role self-evolving paradigms to VLMs is far from straightforward. As shown in Fig. 1 (left), **firstly,** some existing methods (He et al., 2025; Li et al., 2026) typically adopt a coarse-grained role alternation scheme, where the questioner and the solver are optimized in separate long phases. This delayed interaction makes it difficult for question generation and solver adaptation to stay well coordinated, thereby reducing the overall efficiency of the self-evolving loop. **Secondly,** question generation becomes substantially more challenging in the multimodal setting. Unlike in LLM self-evolution, where the model only needs to propose text-based tasks, a VLM must first understand the visual content and then formulate meaningful image-grounded questions. This added difficulty makes self-generated supervision much more prone to degeneration. In our experiments, we observe that without explicit control, self-evolution gradually suffers from both question quality degradation and diversity collapse. Specifically, the generated questions increasingly exhibit semantic errors, pseudo-difficult questions that are weakly grounded in the image, and even invalid questions. Meanwhile, the question distribution progressively collapses toward a few easy-to-generate categories, such as math and counting. Such degeneration weakens the usability and coverage of self-generated supervision, ultimately limiting sustained capability improvement. Therefore, effective self-evolving learning for VLMs requires not only more efficient role interaction, but also mechanisms that improve the reliability of self-generated supervision, especially in terms of question quality and diversity.

To address these issues (Fig. 1 (right)), we first introduce *fine-grained role alternation*, a more online-style training strategy that divides long role-switching phases into shorter update cycles and thereby shortens the feedback loop for role adaptation. On top of this, we further introduce a unified mechanism for question quality and diversity control. Specifically, we employ a *quality supervisor* to constrain question validity and verify pseudo-label reliability, reducing the interference of low-quality questions and noisy samples. We also introduce *skill-aware dynamic balancing* to dynamically regulate the question distribution and alleviate the progressive collapse of generated questions toward a narrow set of modes. In this way, fine-grained alternation improves the timeliness of role coordination, while quality and diversity control ensures the usability of self-generated supervision. Together, they support a more reliable and effective self-evolving process for VLMs.

Our contributions are as follows:

- We identify the key limitations of existing VLM self-evolving methods, including feedback lag caused by coarse-grained role alternation, as well as the progressive degradation of question quality and collapse of diversity during self-evolution.
- We propose a set of reliability-oriented mechanisms for VLM self-evolution, including fine-grained role alternation to reduce feedback lag, a quality supervisor to improve question

validity and pseudo-label reliability, and skill-aware dynamic balancing to mitigate diversity collapse and maintain broad supervision coverage.

- We present **RISE**, a self-evolving framework for vision-language models, which integrates the above designs into a unified training pipeline and enables more reliable and effective improvement from unlabeled images.

## 2 Related Work

**Post-Training for Vision-Language Models.** Recent post-training of vision-language models (VLMs) has evolved from supervised instruction tuning toward reinforcement learning-based optimization. Early works such as LLaVA mainly rely on visual instruction tuning to align visual encoders with language model backbones and improve multimodal instruction-following ability (Li et al., 2024a; Liu et al., 2023; Dai et al., 2026b). More recent R1-style and RLVR-based methods further show that reinforcement learning can enhance multimodal reasoning, visual understanding, and generalization when suitable reward signals are available (Shen et al., 2025; Meng et al., 2025; Deng et al., 2025b; Chen et al., 2025; Chu et al., 2026; Dai et al., 2026a; Hao et al., 2026; Wang et al., 2026). However, most existing post-training methods still depend on human-curated instruction data, annotated reasoning traces, or externally constructed verifiable tasks, which are costly to scale for reasoning-intensive visual learning (Yang et al., 2025; Huang et al., 2025b). This motivates self-evolving VLMs, where models exploit unlabeled images and self-generated supervision to construct their own training signals with minimal human annotation.

**Self-Evolving Learning.** Recent work has shown that large language models can improve themselves with little or no human supervision by exploiting self-generated learning signals, such as pseudo-labels, self-consistency, and verifiable rewards (Zelikman et al., 2022; Huang et al., 2023; Wang et al., 2023; Gulcehre et al., 2023; Singh et al., 2023; Ma et al., 2026; Yang et al., 2026b). Early efforts such as *TTRL* (Zuo et al., 2025) explored how models can learn from self-generated weak supervision without relying on explicit ground-truth labels, while more recent dual-role frameworks, including *R-Zero* (Huang et al., 2025a) and *Absolute Zero* (Zhao et al., 2025), further cast self-evolution as a closed loop between complementary roles that generate and solve tasks. Inspired by these advances in LLMs, recent studies have extended this paradigm to vision-language models, including *VisPlay* (He et al., 2025), *Vision-Zero* (Wang et al., 2025), *EvoLMM* (Thawakar et al., 2025), and *MM-Zero* (Li et al., 2026), showing that multimodal models can also benefit from self-play supervision under unlabeled or weakly supervised settings. However, VLM self-evolution is substantially more fragile, since question generation must remain visually grounded, making the intermediate supervision more prone to quality degradation and diversity collapse. Therefore, rather than further establishing the self-evolving pipeline itself, our work focuses on improving its reliability through better interaction efficiency, question quality control, and skill diversity maintenance.

## 3 Preliminaries

### 3.1 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) is a reinforcement learning (Cao et al., 2024) method that updates a policy by comparing multiple responses generated for the same input, without relying on a separately learned value model. Its key idea is to normalize rewards within a response group and use the resulting relative advantages for policy optimization.

Given an input prompt $p$, the current policy $\pi_{\theta_{\text{old}}}$ generates a group of $G$ responses $\{x_1, \ldots, x_G\}$. Each response $x_i$ is assigned a scalar reward $r_i$. The relative advantage of response $x_i$ is computed as

$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G) + \epsilon_{\text{norm}}}, \tag{1}$$

where $\epsilon_{\text{norm}}$ is a small constant for numerical stability. The policy is then updated with a clipped objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \min\left(\frac{\pi_\theta(x_i)}{\pi_{\theta_{\text{old}}}(x_i)}\hat{A}_i,\ \text{clip}\left(\frac{\pi_\theta(x_i)}{\pi_{\theta_{\text{old}}}(x_i)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right). \tag{2}$$

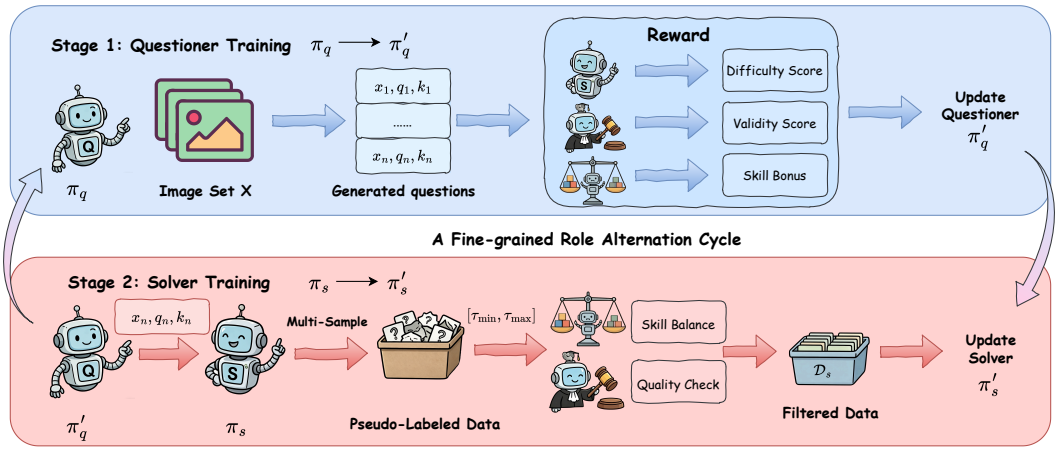


Figure 2: Overview of **RISE**, a dual-role self-evolving framework for VLMs that alternates questioner and solver training with quality supervision and skill balancing.

Such an optimization strategy is well suited to our VLM self-evolving framework, where model-generated answers are directly verifiable and therefore naturally amenable to RLVR.

### 3.2 Dual-Role Self-Evolving Pipeline

Recent self-evolving frameworks for LLMs and VLMs, such as *R-Zero* and *VisPlay*, typically follow a dual-role iterative pipeline. Under this paradigm, the questioner is first optimized according to feedback from the current solver so as to generate questions near the solver's capability boundary. Then, pseudo-labeled training data are constructed from the updated questioner and the majority-voted solver responses, and the solver is subsequently optimized on these data. A conventional dual-role self-evolving pipeline can be written as

$$\pi_q^{(t+1)} = \mathrm{Update}_q^B\big(\pi_q^{(t)}; \pi_s^{(t)}\big),\ \mathcal{D}^{(t)} = \mathrm{Construct}\big(\pi_q^{(t+1)}, \pi_s^{(t)}\big),\ \pi_s^{(t+1)} = \mathrm{Update}_s^B\big(\pi_s^{(t)}; \mathcal{D}^{(t)}\big). \tag{3}$$

where $\pi_q^{(t)}$ and $\pi_s^{(t)}$ denote the questioner and the solver at self-evolving round $t$, respectively; $\mathrm{Update}_q^B$ and $\mathrm{Update}_s^B$ denote applying $B$ successive optimization steps to the questioner and the solver, respectively; and $\mathcal{D}^{(t)} = \{(x, q, \hat{y})\}$ denotes the pseudo-labeled training data constructed at self-evolving round $t$ for solver training.

In this way, question generation and question solving together form a closed self-improvement loop, which serves as the common foundation of existing dual-role self-evolving frameworks.

## 4 Method

### 4.1 Overview

As shown in Fig. 2, given an unlabeled image set $\mathcal{X} = \{x_i\}_{i=1}^N$, we use a single vision-language model to play two functional roles: a *questioner* $\pi_q$ and a *solver* $\pi_s$. Our framework is built under a *fine-grained role alternation* training strategy, which enables more timely feedback exchange between the two roles during self-evolution. Under this strategy, we first train the questioner to generate informative questions for $\mathcal{X}$, using reward signals derived from the solver's response quality and the supervisor feedback. Second, based on the questions generated by the current questioner and the corresponding sampled responses from the solver, we construct and filter pseudo-labeled training data. Third, we train the solver on the filtered pseudo-labeled data to improve visual reasoning capability. The framework is further equipped with two quality and diversity control mechanisms: a quality supervisor to ensure question quality and pseudo-label reliability, and a skill-aware dynamic balancing mechanism to maintain diverse skill coverage during self-evolution.

### 4.2 Fine-Grained Role Alternation

As formalized in Eq. (3), conventional dual-role self-evolution typically relies on *coarse-grained role alternation*: within one alternation round, the questioner keeps adapting to the old solver policy $\pi_s^{(t)}$ for an entire phase of $B$ update steps, while the solver is later trained for another phase of $B$ update steps on data induced by this stale interaction. As a result, the two roles are synchronized only once per round, which slows down capability alignment between them.

To mitigate this lag, we adopt a *fine-grained role alternation* strategy. Instead of switching roles only once after $B$ update steps, we split the same overall update budget into $n$ shorter phases of $b$ steps. We define one short alternation cycle of length $b$ as

$$\pi_q' = \mathrm{Update}_q^b\big(\pi_q; \pi_s\big), \qquad \mathcal{D}' = \mathrm{Construct}\big(\pi_q', \pi_s\big), \qquad \pi_s' = \mathrm{Update}_s^b\big(\pi_s; \mathcal{D}'\big). \tag{4}$$

Then our fine-grained role alternation can be summarized as

$$\big(\pi_q^{(t+1)}, \pi_s^{(t+1)}\big) = \mathrm{Alt}_b^n\big(\pi_q^{(t)}, \pi_s^{(t)}\big), \qquad B = nb, \tag{5}$$

where $\mathrm{Alt}_b$ denotes one short alternation cycle. In this way, the feedback loop between question generation and solver adaptation is substantially shortened: the questioner can receive feedback from a more up-to-date solver after every short phase, and the solver can also be updated using data produced by a more up-to-date questioner, while keeping the same overall update budget. Consequently, the two roles can complete capability alignment more quickly, making the overall self-evolving process more efficient and allowing the model to reach a stronger performance regime earlier.

### 4.3 Quality and Diversity Control

#### 4.3.1 Quality Supervisor

If the questioner is optimized only for generating questions near the solver's capability boundary, it may exploit this objective by producing pseudo-difficult but meaningless questions, such as weakly grounded, semantically ill-posed, skill-mismatched, or visually unanswerable ones. To avoid this issue, we introduce a supervisor to constrain question quality and verify pseudo-label reliability.

Instead of training an additional judge model, we let the supervisor share parameters with the current solver, i.e., $\pi_{\mathrm{sup}} \equiv \pi_s$. This shared-parameter design introduces no extra model cost and also improves internal consistency: a model that answers questions should also be able to assess whether those questions and the generated answers are acceptable. Meanwhile, we do not let the questioner itself act as the supervisor, which helps reduce reward hacking by avoiding self-judging.

Given an image $x$, the questioner generates a structured output $(k, q) \sim \pi_q(\cdot \mid x)$, where $k$ denotes the declared skill and $q$ is the question text. The supervisor serves two purposes in our framework. First, it participates in questioner training by judging whether a generated question is valid. We use a binary signal $v(x, q, k) \in \{0, 1\}$, which equals 1 only if the question is image-grounded, semantically well-formed, answerable from the image alone, and consistent with its declared skill. This signal is directly incorporated into the questioner reward so that the model cannot obtain high reward merely by confusing the solver.

Second, it is used during pseudo-label data construction to verify whether the candidate answer is acceptable. We define another binary signal $u(x, q, \hat{y}) \in \{0, 1\}$ to verify the correctness of the majority-voted answer $\hat{y}$ produced by the solver.

In practice, both $v(x, q, k)$ and $u(x, q, \hat{y})$ are obtained by prompt-based judgment using the supervisor.

#### 4.3.2 Skill-Aware Dynamic Balancing

During self-evolution, the questioner tends to gradually favor a few skills that yield higher reward more easily, such as counting or math. Such mode collapse makes the generated questions increasingly narrow and limits the range of reasoning skills the model can acquire.

To address this issue, we introduce a skill-aware dynamic balancing mechanism. The key idea is to maintain the recent frequency of each skill during training and provide additional encouragement to under-represented skills, thereby promoting more diverse question generation.

We define the skill set $\mathcal{K}$ as comprising six categories: coarse perception, fine-grained perception, instance reasoning, logical reasoning, math & counting, and science & technology. For each skill $k \in \mathcal{K}$, let $n_k$ denote the recent number of generated questions whose declared skill is $k$ before the current alternation cycle. We compute the average skill frequency as

$$\bar{n} = \frac{1}{|\mathcal{K}|} \sum_{k' \in \mathcal{K}} n_{k'}. \tag{6}$$

For a question with declared skill $k$, we define the skill bonus as

$$b_{\text{skill}}(k) = \max\left(\frac{\bar{n} - n_k}{\bar{n}}, 0\right). \tag{7}$$

In this way, only relatively under-represented skills receive additional reward, while dominant skills receive no extra encouragement. Combined with fine-grained role alternation, it also enables more timely adjustment of the question skill distribution during self-evolution.

We further apply skill-stratified sampling when constructing the final pseudo-labeled data, by drawing approximately equal-sized subsets from the six skill categories, so that skill balancing affects both question generation and solver training.

### 4.4 Training Objectives

#### 4.4.1 Questioner Objective

The questioner is encouraged to generate questions near the solver's capability boundary. To assess this, we first let the solver sample $M$ responses:

$$y_j \sim \pi_s(\cdot \mid x, q), \qquad j = 1, \ldots, M. \tag{8}$$

Based on these responses, we compute the self-consistency score

$$c(x, q) = \frac{1}{M} \max_a \sum_{j=1}^{M} \mathbf{1}[y_j = a]. \tag{9}$$

A larger $c(x, q)$ indicates that the solver responses are more stable for the question $(x, q)$. We use this score to construct the difficulty score (Sun et al., 2025)

$$d(x, q) = \min\big(c(x, q),\, 1 - c(x, q)\big), \tag{10}$$

which favors questions near the solver's current capability boundary rather than overly easy or highly unstable ones.

The questioner reward is defined as

$$r_q(x, q, k) = \begin{cases} -1, & \text{invalid format}, \\ d(x, q) + \lambda_v\, v(x, q, k) + \lambda_s\, b_{\text{skill}}(k), & \text{valid format}. \end{cases} \tag{11}$$

which encourages the questioner to generate questions that are moderately difficult, semantically valid, and diverse in skill coverage.

#### 4.4.2 Solver Objective

Based on the questions generated by the questioner trained in the current alternation round and the corresponding sampled solver responses, we construct solver training data. For each candidate question, the pseudo label is obtained by majority voting over the sampled solver outputs:

$$\hat{y} = \arg\max_a \sum_{j=1}^{M} \mathbf{1}[y_j = a]. \tag{12}$$

We then reuse the same self-consistency score $c(x, q)$ to estimate sample confidence, and pre-filter candidate samples by

$$c(x, q) \in [\tau_{\min}, \tau_{\max}], \tag{13}$$

where overly easy questions usually provide limited learning value, while highly unstable ones are often noisy. After this confidence-based pre-filtering, we further retain only samples satisfying

$$v(x, q, k) = 1, \qquad u(x, q, \hat{y}) = 1. \tag{14}$$

We then apply skill-balanced sampling over the remaining samples to form the final pseudo-labeled training set, which is defined as

$$\mathcal{D}_s = \{(x, q, \hat{y}, k)\}. \tag{15}$$

The solver is trained on the filtered pseudo-labeled training set $\mathcal{D}_s$. Given a sample $(x, q, \hat{y}, k)$ and solver output $\tilde{y}$, the solver reward is defined as

$$r_s(\tilde{y}, \hat{y}) = \begin{cases} -1, & \text{invalid format,} \\ \mathbf{1}[\text{extract}(\tilde{y}) = \hat{y}], & \text{valid format.} \end{cases} \tag{16}$$

# 5 Experiments

## 5.1 Experimental Setup

**Models and Image Set.** To validate the effectiveness of the proposed self-evolving framework, we conduct experiments on two vision-language models of different scales and generations, Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B (Bai et al., 2025a). In our experiments, we initialize the questioner and the solver from the same base model, and train them under the same self-evolving pipeline. For self-evolving training, we use the unlabeled image collection from Vision-47K (Feng et al., 2025; Li et al., 2025), which contains about 47K images and covers diverse visual scenes and task sources, providing a diverse image pool.

**Training Details.** Following the setting in prior coarse-grained dual-role self-evolving frameworks, we use a total update budget of $B = 20$ as the reference within one alternation round. For fair comparison, our fine-grained role alternation keeps the same total budget and sets $n = 4$ and $b = 5$, such that $B = nb = 20$. In each update step, we sample 256 unlabeled images for self-evolving training. For each question generated by the questioner, the solver samples $M = 10$ responses for difficulty estimation and pseudo-label construction. For pseudo-label filtering, we set the self-consistency threshold range to $[\tau_{\min}, \tau_{\max}] = [0.3, 0.8]$. In the questioner reward in Eq. (11), we set the weights of the validity term and the skill bonus term to $\lambda_v = 0.2$ and $\lambda_s = 0.2$, respectively.

**Evaluation Benchmarks and Protocol.** We evaluate the evolved models on MMMU (Yue et al., 2024), MMVet (Yu et al., 2023), RealWorldQA (xAI, 2024), ChartQA (Masry et al., 2022), MathVista (Lu et al., 2023), MathVerse (Zhang et al., 2024b), and MathVision (Wang et al., 2024). These benchmarks jointly cover a broad range of capabilities, including general visual understanding, chart understanding, and multimodal mathematical reasoning, thereby providing a comprehensive testbed for assessing whether self-evolution improves both general-purpose and reasoning-intensive visual capabilities. We compare the models before and after self-evolution under the same inference setting on all evaluation benchmarks. Following an LLM-as-a-Judge protocol (Gu et al., 2024; Li et al., 2024b), we use Qwen3-Max to assess answer correctness, as it better handles semantically equivalent answers than exact string matching. In addition, we compute an overall score as the sample-size weighted average over all benchmarks. More details are provided in Appendix A.

## 5.2 Main Results

**Self-Evolving Results.** Table 1 summarizes the self-evolving results of RISE on Qwen2.5-VL-7B-Instruct and Qwen3-VL-8B-Instruct. RISE consistently improves both backbones across self-evolving steps. On Qwen2.5-VL-7B-Instruct, the weighted average score increases from 48.17 to 51.20, 51.74, and 52.26 after 20, 40, and 60 steps, respectively, yielding a total gain of +4.09. On Qwen3-VL-8B-Instruct, the average score improves from 53.38 to 55.17, 56.15, and 57.37, corresponding to a total gain of +3.99 after 60 steps. These gains are broad across evaluation domains, including general visual understanding, chart understanding, and multimodal mathematical reasoning, with particularly strong improvements on reasoning-intensive benchmarks such as MathVerse and MathVision. We attribute these stable improvements to the explicit control of question quality and skill distribution

Table 1: Main results of RISE on two base VLMs under different self-evolving steps. We report accuracy on seven benchmarks and the sample-size weighted average (AVG).

| Model Name | MMMU | MMVet | RealWQA | ChartQA | MathVerse | MathVision | MathVista | AVG |
|---|---|---|---|---|---|---|---|---|
| **Qwen2.5-VL-7B-Instruct** | | | | | | | | |
| Base Model | 49.71 | 51.83 | 57.12 | 79.48 | 40.48 | 23.21 | 58.70 | 48.17 |
| RISE-Step20 | 54.03 | **57.80** | 57.52 | 81.68 | 43.65 | 27.36 | 60.40 | 51.20 |
| RISE-Step40 | 54.03 | 56.88 | 58.30 | 81.84 | 44.19 | **28.73** | 60.10 | 51.74 |
| RISE-Step60 | **55.31** | **57.80** | **60.39** | **82.08** | **44.95** | 28.32 | **61.10** | **52.26** |
| **Qwen3-VL-8B-Instruct** | | | | | | | | |
| Base Model | 55.31 | 60.55 | 67.45 | 81.44 | 46.75 | 27.32 | 65.80 | 53.38 |
| RISE-Step20 | 55.54 | 61.93 | 68.50 | 82.28 | 48.65 | 31.28 | 65.60 | 55.17 |
| RISE-Step40 | 58.11 | 62.39 | **70.85** | 82.24 | 49.06 | 32.80 | 67.70 | 56.15 |
| RISE-Step60 | **59.63** | **62.84** | 70.33 | **83.64** | **49.11** | **36.10** | **68.70** | **57.37** |

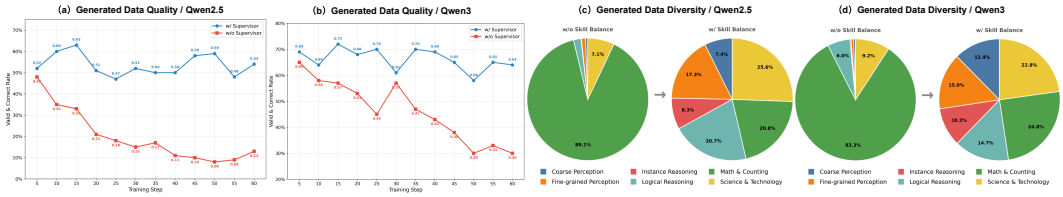


Figure 3: Statistics of question quality and diversity during self-evolution.

during self-evolution, which provides more reliable and learning-effective supervision throughout the intermediate training process. We also observe that, on most benchmarks, the marginal gains become smaller in later stages, suggesting a gradual convergence effect. From a methodological perspective, the solver is updated by using pseudo labels constructed from majority-voted responses, i.e., a maj@$n$ policy, to improve the current policy. While this form of supervision is effective in correcting evident errors at earlier stages, its incremental benefit naturally diminishes as the solver becomes stronger. As a result, self-evolution continues to yield positive gains in later stages, but the improvement typically becomes more moderate over time.

**Analysis of the Quality and Diversity.** We conduct a statistical analysis of the intermediate VQA data generated during self-evolution, as shown in Fig. 3, to verify the degeneration phenomena and the effectiveness of our method. For **question quality**, we evaluate the validity and correctness of generated questions. Without a supervisor, the valid-and-correct rate declines steadily and drops below 10% in later stages, indicating the accumulation of low-quality questions. In contrast, introducing the supervisor keeps question quality consistently high, showing that it effectively prevents degradation and ensures reliable intermediate supervision. For **question diversity**, the skill distribution shows that without skill balance, generated questions collapse toward a few high-reward skills. This is mainly because the questioner reward includes the difficulty score, which gradually drives the model to favor skill types that are easier to obtain high difficulty rewards from, thereby narrowing the coverage of training signals. With skill-aware dynamic balancing, the skill distribution becomes more balanced, effectively alleviating mode collapse. These two phenomena can be consistently observed during the self-evolution of both base models. Additional qualitative examples and statistics are provided in Appendix B, and the reliability of the supervisor judgment is further analyzed in Appendix C.1.

### 5.3 Ablation Study on RISE Components

Table 2 reports the ablation results of RISE on Qwen2.5-VL-7B-Instruct. We analyze the effects of the three core components in RISE, and summarize four main observations. Additional ablation study and analysis are provided in Appendix C.

**(1) Fine-grained role alternation improves self-evolving efficiency.** Regardless of whether the quality supervisor is used, fine-grained role alternation generally reaches a stronger performance regime with fewer update steps than its coarse-grained counterpart. For example, without the

Table 2: Ablation study of RISE on Qwen2.5-VL-7B-Instruct. "Fine-grained" and "Coarse-grained" denote the role alternation strategy, and "Sup." denotes the quality supervisor. All non-RISE variants remove the skill balancing module, while RISE denotes the full model.

| Step | Variant | MMMU | MMVet | RealWQA | ChartQA | MathVerse | MathVision | MathVista | AVG |
|---|---|---|---|---|---|---|---|---|---|
| – | Base Model | 49.71 | 51.83 | 57.12 | 79.48 | 40.48 | 23.21 | 58.70 | 48.17 |
| Step 20 | Fine-grained w/o Sup. | 51.58 | **59.63** | 57.65 | **81.68** | 42.89 | 26.80 | 57.30 | 50.43 |
| | Coarse-grained w/o Sup. | 52.51 | 56.42 | 58.17 | 80.76 | 41.83 | 24.66 | 58.90 | 49.58 |
| | Fine-grained + Sup. | 53.68 | 57.80 | 58.04 | 81.08 | 42.69 | 27.03 | 59.80 | 50.64 |
| | Coarse-grained + Sup. | 51.23 | 58.26 | **58.56** | 81.56 | 41.83 | 24.51 | **60.40** | 49.81 |
| | **RISE** | **54.03** | 57.80 | 57.52 | **81.68** | **43.65** | **27.36** | **60.40** | **51.20** |
| Step 40 | Fine-grained w/o Sup. | 52.51 | 56.42 | 57.78 | 81.72 | 42.39 | 26.40 | 60.00 | 50.42 |
| | Coarse-grained w/o Sup. | 52.74 | 56.42 | 59.35 | 81.04 | 43.48 | 24.99 | 58.60 | 50.32 |
| | Fine-grained + Sup. | 52.86 | **60.09** | 58.82 | 81.64 | **45.15** | 27.21 | 59.90 | 51.65 |
| | Coarse-grained + Sup. | 51.81 | **60.09** | **59.87** | 81.44 | 43.93 | 24.92 | **61.30** | 50.80 |
| | **RISE** | **54.03** | 56.88 | 58.30 | **81.84** | 44.19 | **28.73** | 60.10 | **51.74** |
| Step 60 | Fine-grained w/o Sup. | 41.19 | 51.38 | 55.56 | 80.00 | 33.83 | 21.62 | 56.60 | 44.85 |
| | Coarse-grained w/o Sup. | 53.09 | 56.42 | 59.08 | 81.44 | 43.10 | 22.81 | 60.50 | 49.96 |
| | Fine-grained + Sup. | 53.56 | 56.42 | 57.25 | 82.00 | **45.18** | 27.51 | 60.20 | 51.71 |
| | Coarse-grained + Sup. | 54.38 | 57.34 | 58.82 | 81.48 | 44.67 | 23.99 | 60.00 | 50.80 |
| | **RISE** | **55.31** | **57.80** | **60.39** | **82.08** | 44.95 | **28.32** | **61.10** | **52.26** |

supervisor, *Fine-grained w/o Sup.* already achieves an AVG of 50.43 at step 20, which is higher than *Coarse-grained w/o Sup.* at step 40 (50.32) and step 60 (49.96). A similar trend is also observed when the supervisor is enabled. These results show that shortening the feedback loop between the questioner and the solver makes self-evolution more efficient. We further analyze the effect of different alternation granularities in Appendix C.3.

**(2) Faster alternation alone can aggravate quality degradation.** Although fine-grained alternation is more efficient, it is not sufficient by itself to ensure stable evolution. Without the quality supervisor, the AVG of *Fine-grained w/o Sup.* drops from 50.43 to 50.42, and then sharply to 44.85 at step 60. This suggests that when role interaction becomes faster but question quality is not explicitly constrained, the questioner is more likely to exploit the reward and drift toward low-quality or reward-hacked questions, which eventually harms pseudo-label quality and solver training.

**(3) The supervisor enables fast yet stable evolution.** Comparing *Fine-grained + Sup.* with *Fine-grained w/o Sup.* shows that the supervisor effectively stabilizes the fine-grained self-evolving process. With the supervisor, the AVG remains strong from 50.64 to 51.71, avoiding the severe collapse observed in the unsupervised fine-grained setting. Similar benefits can also be observed in the coarse-grained setting. This result is consistent with our design motivation: the supervisor helps enforce question validity and pseudo-label reliability, thereby allowing fine-grained role alternation to retain its efficiency advantage without sacrificing training stability.

**(4) Skill balancing further improves diversity and final performance.** On top of *Fine-grained + Sup.* setting, adding skill balancing gives the full RISE model and yields the best overall performance. RISE brings further gains across steps, and these gains are broad across benchmarks rather than coming from a single dataset. This indicates that skill balancing helps prevent question generation from collapsing to a narrow set of easy-reward skills and supports more balanced capability growth.

## 6 Conclusion

In this paper, we present **RISE**, a reliable self-evolving framework for vision-language models under unlabeled images. By identifying the key limitations of existing methods, including delayed role interaction, question quality degradation, and diversity collapse, we develop a unified solution that combines fine-grained role alternation, a quality supervisor, and skill-aware dynamic balancing. These components work together to improve interaction efficiency, ensure the reliability of self-generated supervision, and maintain diverse skill coverage throughout the evolution process. Extensive experiments show that RISE consistently brings stable and sustained performance gains across different model scales and multiple benchmarks. Overall, our results suggest that reliable interaction and controlled supervision are crucial for effective and reliable VLM self-evolution.

## Reproducibility Statement

To ensure the reproducibility of our work, we provide detailed experimental settings and additional analyses in the appendix. Appendix A provides the training hyperparameters, prompt templates, and training algorithm needed to reproduce our experiments. Appendix B illustrates the quality, diversity, and evolution of the generated data through examples and statistics. Appendix C provides extensive additional experiments and analyses, covering supervisor reliability (Appendix C.1), self-evolution with imperfect pseudo-labels (Appendix C.2), ablations on role alternation frequency (Appendix C.3) and on Qwen3-VL (Appendix C.4), performance stability and variability across independent runs (Appendix C.5), supplementary experiments on evaluation protocols (Appendix C.6), performance over longer self-evolution (Appendix C.7), computational cost (Appendix C.8), and reference comparisons with prior methods (Appendix C.9). Code will also be made available.

## AI Use Statement

In preparing this manuscript, we used large language models mainly for language polishing and translation. The authors take responsibility for the final content.

## References

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.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2024.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*, 2025.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=inccdtfx8x.

Yanqi Dai, Yuxiang Ji, Xiao Zhang, Yong Wang, Xiangxiang Chu, and Zhiwu Lu. Harder is better: Boosting mathematical reasoning via difficulty-aware grpo and multi-aspect question reformulation. *arXiv preprint arXiv:2601.20614*, 2026a.

Yanqi Dai, Yong Wang, Zebin You, Dong Jing, Xiangxiang Chu, and Zhiwu Lu. Cotbal: Comprehensive task balancing for multi-task visual instruction tuning. *arXiv preprint arXiv:2403.04343*, 2026b.

Shijian Deng, Kai Wang, Tianyu Yang, Harsh Singh, and Yapeng Tian. Self-improvement in multimodal large language models: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1987–2006, 2025a.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv e-prints*, pp. arXiv–2503, 2025b.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

Aiming Hao, Chen Zhu, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. Ranking-aware reinforcement learning for ordinal ranking. *arXiv preprint arXiv:2601.20585*, 2026.

Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. Visplay: Self-evolving vision-language models from images. *arXiv preprint arXiv:2511.15661*, 2025.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025a.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 1051–1068, 2023.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025b.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Chen Li, Yixiao Ge, Dian Li, and Ying Shan. Vision-language instruction tuning: A review and analysis. *arXiv preprint arXiv:2311.08172*, 2023.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024b.

Ming Li, Keyu Chen, Ziqian Bi, Ming Liu, Xinyuan Song, Zekun Jiang, Tianyang Wang, Benji Peng, Qian Niu, Junyu Liu, et al. Surveying the mllm landscape: A meta-review of current surveys. *arXiv preprint arXiv:2409.18991*, 2024c.

Zongxia Li, Wenhao Yu, Chengsong Huang, Rui Liu, Zhenwen Liang, Fuxiao Liu, Jingxi Che, Dian Yu, Jordan Boyd-Graber, Haitao Mi, et al. Self-rewarding vision-language model via reasoning decomposition. *arXiv preprint arXiv:2508.19652*, 2025.

Zongxia Li, Hongyang Du, Chengsong Huang, Xiyang Wu, Lantao Yu, Yicheng He, Jing Xie, Xiaomin Wu, Zhichao Liu, Jiarui Zhang, et al. Mm-zero: Self-evolving multi-model vision language models from zero data. *arXiv preprint arXiv:2603.09206*, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. Spice: Self-play in corpus environments improves reasoning. *arXiv preprint arXiv:2510.24684*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

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.

Ziyu Ma, Shidong Yang, Yuxiang Ji, Xucong Wang, Yong Wang, Yiming Hu, Tongwen Huang, and Xiangxiang Chu. Skillclaw: Let skills evolve collectively with agentic evolver. *arXiv preprint arXiv:2604.08377*, 2026.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

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.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models, 2024.

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.

Omkar Thawakar, Shravan Venkatraman, Ritesh Thawkar, Abdelrahman Shaker, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Khan. Evolmm: Self-evolving large multimodal models with continuous rewards. *arXiv preprint arXiv:2511.16672*, 2025.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37: 87310–87356, 2024.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

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.

Qinsi Wang, Bo Liu, Tianyi Zhou, Jing Shi, Yueqian Lin, Yiran Chen, Hai Helen Li, Kun Wan, and Wentian Zhao. Vision-zero: Scalable vlm self-improvement via strategic gamified self-play. *arXiv preprint arXiv:2509.25541*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Zengbin Wang, Feng Xiong, Liang Lin, Xuecai Hu, Yong Wang, Yanlin Wang, Man Zhang, and Xiangxiang Chu. Visually-guided policy optimization for multimodal reasoning. *arXiv preprint arXiv:2604.09349*, 2026.

xAI. Realworldqa: Real-world spatial understanding benchmark. https://huggingface.co/datasets/xai-org/RealworldQA, 2024. CC BY-ND 4.0 license. Benchmark dataset released with Grok-1.5 Vision Preview; see also https://x.ai/news/grok-1.5v.

Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning. *arXiv preprint arXiv:2511.16043*, 2025.

Shidong Yang, Tongwen Huang, Hao Wen, Yong Wang, Li Chen, and Xiangxiang Chu. Entropy-guided data-efficient training for multimodal reasoning reward models. *arXiv preprint arXiv:2602.01884*, 2026a.

Shidong Yang, Ziyu Ma, Tongwen Huang, Yiming Hu, Yong Wang, and Xiangxiang Chu. Coevolve: Training llm agents via agent-data mutual evolution. *arXiv preprint arXiv:2604.15840*, 2026b.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2376–2385, 2025.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Wenhao Yu, Zhenwen Liang, Chengsong Huang, Kishan Panaganti, Tianqing Fang, Haitao Mi, and Dong Yu. Guided self-evolving llms with minimal human supervision. *arXiv preprint arXiv:2512.02472*, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

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

Jiarui Zhang, Jinyi Hu, Mahyar Khayatkhoei, Filip Ilievski, and Maosong Sun. Exploring perceptual limitation of multimodal large language models. *arXiv preprint arXiv:2402.07384*, 2024a.

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*, pp. 169–186. Springer, 2024b.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025.

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.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

# APPENDIX

## A DETAILED TRAINING AND IMPLEMENTATION DETAILS

### A.1 HYPERPARAMETER SETTINGS

We use the same training hyperparameters for the questioner and the solver unless otherwise specified. In our fine-grained role alternation strategy, each short alternation cycle contains $b = 5$ update steps. The whole self-evolving process contains 12 fine-grained alternation cycles. Therefore, the full training process contains 60 update steps in total, and we report the intermediate checkpoints after 20, 40, and 60 update steps.

For GRPO optimization, we use 8 rollouts per training prompt. During question evaluation and pseudo-label construction, the solver samples $M = 10$ responses for each generated question, which are used to estimate self-consistency, compute the difficulty score, and construct majority-voted pseudo-labels. Following the main paper, the reward weights for the validity score and the skill bonus are both set to $\lambda_v = 0.2$ and $\lambda_s = 0.2$. The batch size is set to 256, which means that training for 20 update steps requires generating approximately 5K VQA samples. During rollout generation, we use a temperature of 1.0 and a top-$p$ value of 0.99.

For benchmark evaluation, we use a unified LLM-as-a-Judge protocol to assess answer correctness. This is because model responses to multimodal benchmarks can have diverse surface forms, and several benchmarks include open-ended VQA questions whose correct answers cannot be reliably evaluated by exact string matching alone. Judge-based evaluation is therefore more suitable for recognizing semantically equivalent answers expressed in different forms. For each test sample, we generate one response using greedy decoding, and then use the judge model to determine whether the extracted answer matches the ground-truth answer.

All experiments are conducted on 8 NVIDIA H20 GPUs. Under the matched setting in Appendix C.8, the complete RISE run takes 73 hours, including data generation and filtering.

### A.2 PROMPT TEMPLATES

We use four types of prompts in RISE: question generation, solver answering, question validity judgment, and answer verification. For readability, we omit the model-specific chat template tokens and denote the input image as `[IMAGE]`. The placeholders `{QUESTION}`, `{DECLARED_SKILL}`, `{SKILL_CONTEXT}`, and `{CANDIDATE_ANSWER}` are dynamically filled during training.

**Questioner prompt.** The questioner is instructed to generate one challenging visual reasoning question from the given image. It must declare one skill category and one question type, and output the result in a structured format.

```
System:
You are an intelligent Question Generator. Given an image,
 ↪  generate exactly
one difficult visual reasoning question that is directly
 ↪  grounded in the image.

The question must:
1. require visual analysis or reasoning rather than simple
 ↪  description;
2. belong to exactly one skill category:
coarse perception, fine-grained perception, instance reasoning,
logical reasoning, math & counting, or science & technology;
3. belong to exactly one question type:
multiple choice, numerical, or regression;
4. have a short, unique, and verifiable answer.

Output strictly in the following format:
<skill>...</skill>
```

```
<type>...</type>
<question>...</question>

User:
[IMAGE]
Generate one new challenging reasoning question based on this
↪ image.
```

**Solver prompt.** The solver is prompted to answer the generated question based on the image. To make answer extraction reliable, the final answer is required to appear inside `\boxed{}`.

```
System:
You are a helpful visual reasoning assistant.

User:
[IMAGE]
Please reason step by step carefully based on the image for the
↪ following
question:
{QUESTION}

After completing your reasoning, output the final clean and
↪ concise answer
strictly inside \boxed{}.
```

**Question validity and skill-matching prompt.** The quality supervisor judges whether a generated question is valid and whether the declared skill matches the question. It does not solve the question, but only returns a binary decision.

```
System:
You are a strict visual question validity judge. Decide whether
↪ the question
can be answered solely from the provided image and whether the
↪ declared skill
matches the question.

Use the following skill definition and restriction:
{SKILL_CONTEXT}

Output \boxed{1} only if the question is image-grounded,
↪ well-posed,
answerable from the image alone, and consistent with the
↪ declared skill.
Otherwise, output \boxed{0}.

User:
[IMAGE]
Question: {QUESTION}
Declared Skill: {DECLARED_SKILL}

Judge whether the question is valid and whether the declared
↪ skill is correct.
Do not solve the question. End with \boxed{1} or \boxed{0}.
```

**Answer verification prompt.** After majority voting over solver responses, the supervisor further verifies whether the candidate pseudo-label is correct.

```
System:
You are a strict visual question answering verifier. Given an
↪ image, a question,
```

```
and a candidate answer, decide whether the candidate answer is
↪ correct.
Output \boxed{1} only if the candidate answer is correct.

User:
[IMAGE]
Question: {QUESTION}
Candidate Answer: {CANDIDATE_ANSWER}

Briefly judge the answer correctness and end with \boxed{1} or
↪ \boxed{0}.
```

**Skill context.** For the skill-matching prompt, `{SKILL_CONTEXT}` is dynamically replaced by the definition of the declared skill. We use six skill categories: coarse perception, fine-grained perception, instance reasoning, logical reasoning, math & counting, and science & technology. For non-math skills, we explicitly forbid questions whose main solution path depends on counting or numerical estimation, so as to reduce skill-label leakage toward the math & counting category.

### A.3 Overall Training Algorithm

**Require:** Unlabeled image set $\mathcal{X}$, initial VLM $\pi_0$, skill set $\mathcal{K}$, cycle length $b$, number of cycles $C$
**Ensure:** Evolved solver $\pi_s$

Initialize questioner and solver with the same VLM: $\pi_q \leftarrow \pi_0$, $\pi_s \leftarrow \pi_0$
Set the quality supervisor as $\pi_{\text{sup}} \equiv \pi_s$
Initialize recent skill statistics $\{n_k\}_{k \in \mathcal{K}}$
**for** $c = 1$ to $C$ **do**
  *// Questioner update*
  **for** $t = 1$ to $b$ **do**
    Sample images from $\mathcal{X}$ and generate questions $(q, k)$ with $\pi_q$
    Estimate question difficulty by sampling multiple responses from $\pi_s$
    Use $\pi_{\text{sup}}$ to judge question validity and skill consistency
    Compute the questioner reward with difficulty score, validity score, and skill bonus
    Update $\pi_q$ with GRPO
  **end for**
  *// Pseudo-label construction*
  Generate candidate questions using the updated questioner $\pi_q$
  Sample multiple solver responses and construct majority-voted pseudo-labels
  Filter samples by self-consistency, question validity, and answer verification
  Apply skill-balanced sampling to obtain the solver training set $\mathcal{D}_s$
  Update recent skill statistics $\{n_k\}_{k \in \mathcal{K}}$
  *// Solver update*
  **for** $t = 1$ to $b$ **do**
    Sample pseudo-labeled data $(x, q, \hat{y}, k)$ from $\mathcal{D}_s$
    Compute solver rewards by comparing extracted answers with $\hat{y}$
    Update $\pi_s$ with GRPO
  **end for**
  Refresh the shared supervisor: $\pi_{\text{sup}} \equiv \pi_s$
**end for**
**return** $\pi_s$

Algorithm above provides the overall training procedure of RISE. We initialize the questioner and solver from the same VLM and use the current solver as a quality supervisor. Each fine-grained alternation cycle consists of three stages: questioner update, pseudo-label construction, and solver update. The questioner is optimized with difficulty, validity, and skill-balancing rewards; the generated VQA samples are then filtered by self-consistency and supervisor verification; and the solver is updated on the resulting skill-balanced pseudo-labeled data. This repeated interaction enables more timely role coordination and more reliable self-evolution.

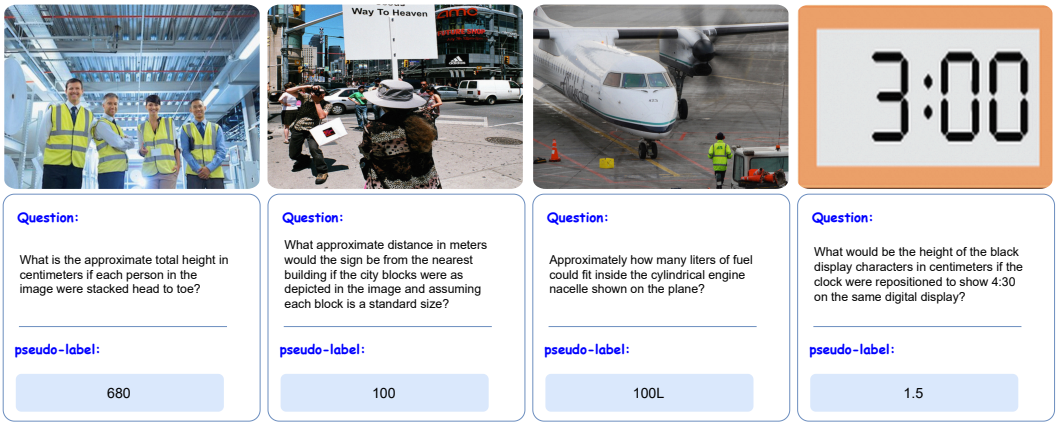


Figure 4: Representative bad cases generated without explicit control of question quality and diversity. Many questions are weakly grounded, ambiguous, or unsuitable as effective supervision. Moreover, the generated content shows a clear collapse toward estimation- and quantity-related questions, reflecting the tendency of difficulty-only optimization to favor uncertain numerical questions.

# B Qualitative Visualization

In this section, we provide additional qualitative results to further illustrate the failure modes of uncontrolled self-evolution and the behavior of RISE during training. Specifically, we present: (1) representative bad cases without quality and diversity control, (2) the evolution of generated questions for the same image across training stages, (3) examples of questions generated by RISE for different skill types, and (4) the skill distribution statistics of the pseudo-labeled VQA data constructed at different training steps.

## B.1 Bad Cases without Quality and Diversity Control

Figure 4 shows representative questions generated without explicit control over question quality and diversity. Two observations can be made. First, many generated questions are of poor quality: some are weakly grounded, some are ambiguous or unverifiable, and some are simply unsuitable as effective VQA supervision. In particular, several questions require arbitrary estimation of quantities or physical values from a single image, making them unreliable as training targets. Second, the generated content exhibits a clear collapse in diversity. Instead of covering a broad range of visual skills, the questions become heavily concentrated on numerical estimation or quantity-related reasoning.

This phenomenon is closely related to the difficulty-driven reward used for the questioner. Since the questioner is encouraged to maximize the difficulty score induced by the solver's self-consistency, its objective is effectively to produce questions whose answer consistency is close to 0.5. In practice, estimation-style questions are a convenient shortcut to achieve this target, because they are often uncertain, weakly constrained, and naturally lead to dispersed solver responses. As a result, without additional quality supervision and skill-aware balancing, self-evolution tends to produce low-quality yet reward-seeking questions and gradually collapses to a narrow subset of skills.

## B.2 Question Evolution on the Same Image

Figure 5 visualizes the questions generated for the same image at different training stages. As training proceeds, the generated questions become progressively more challenging: early-stage questions mainly focus on direct scene understanding or simple recognition, while later-stage questions require more fine-grained analysis, compositional reasoning, or more difficult numerical judgment. This progression indicates that the questioner gradually learns to construct more informative supervision as self-evolution advances.

At the same time, unlike uncontrolled self-evolution, the quality and diversity of generated questions do not collapse. Thanks to the quality supervisor, later questions remain image-grounded, answerable,

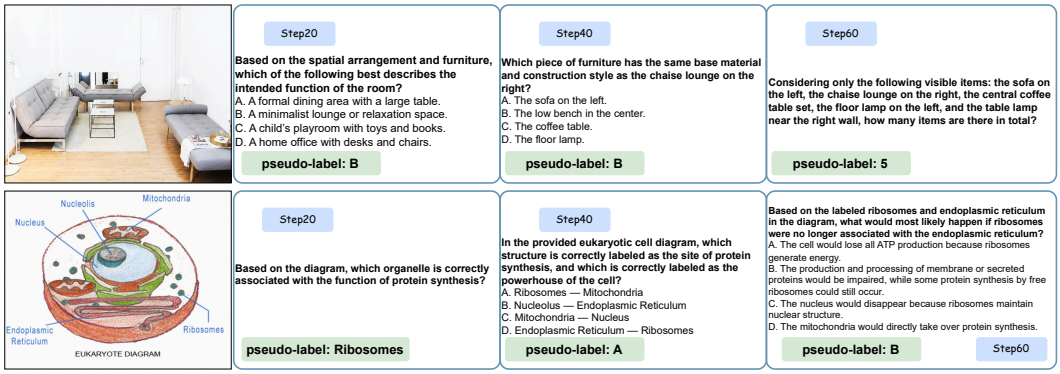


Figure 5: Evolution of generated questions for the same image across different training stages. The generated questions become progressively more challenging over time, while remaining image-grounded and diverse under the guidance of the quality supervisor and skill-aware balancing.

and semantically valid. Meanwhile, skill-aware dynamic balancing prevents the generation process from collapsing to a single high-reward skill and encourages the questioner to maintain diverse question types during training. Therefore, RISE can increase question difficulty while preserving both question quality and skill diversity.

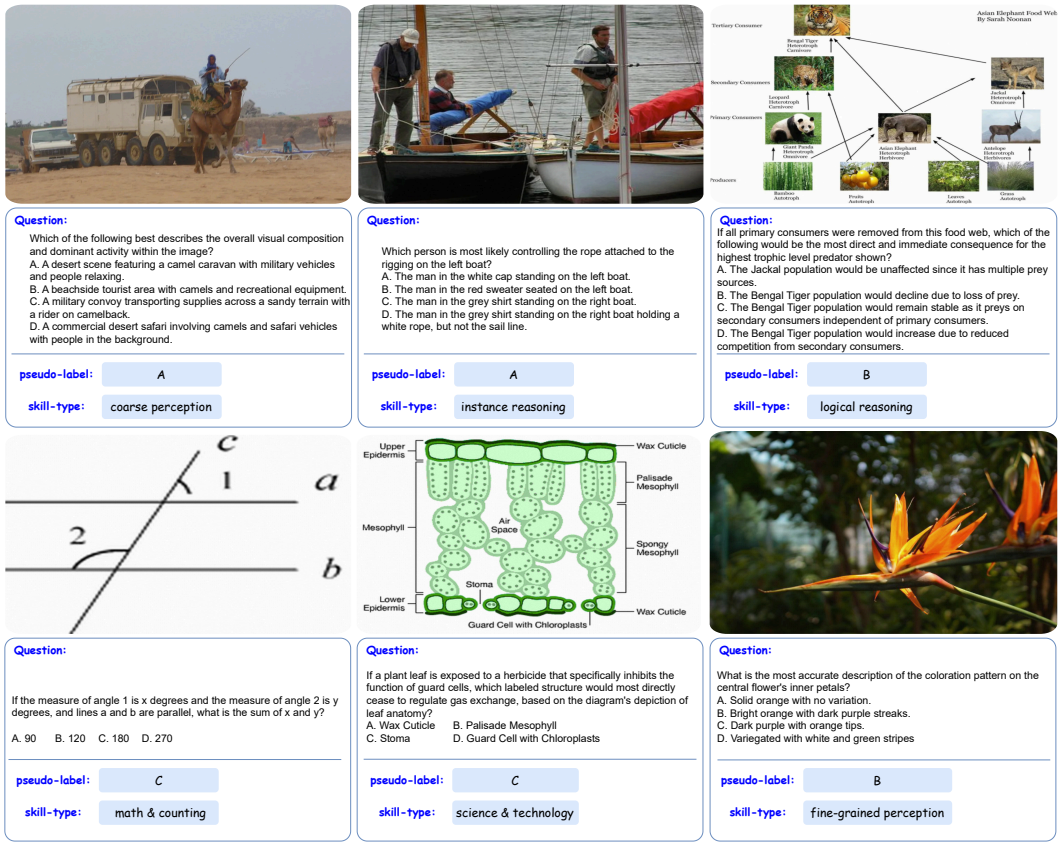


Figure 6: Representative questions generated by RISE for different skill types. The examples show that RISE maintains broad skill coverage, including coarse perception, fine-grained perception, instance reasoning, logical reasoning, math & counting, and science & technology.

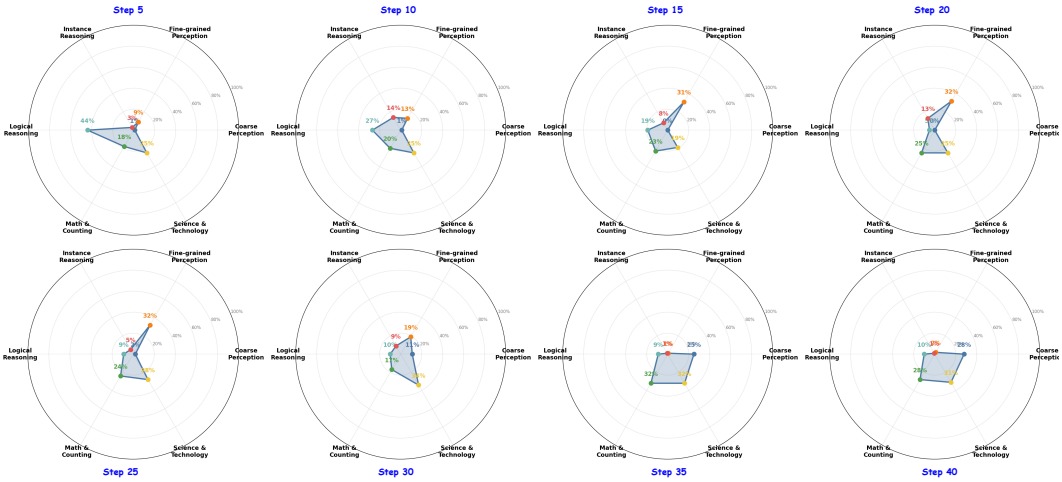


Figure 7: Skill distribution statistics of the pseudo-labeled VQA data constructed by RISE every five training steps. The distributions remain multi-modal rather than collapsing to a single dominant skill, while their relative emphasis changes across stages. Since each round uses 256 randomly sampled images, the observed fluctuations also reflect the interaction between dynamic skill regulation and the sampled image content.

### B.3 Examples of Generated Questions across Skills

Figure 6 presents representative examples of questions generated by RISE for different skill types. These examples cover multiple categories, including coarse perception, fine-grained perception, instance reasoning, logical reasoning, math & counting, and science & technology. The examples qualitatively demonstrate that RISE does not merely generate harder questions, but also maintains broad coverage over different visual and reasoning abilities. This multi-skill coverage is important for preventing mode collapse and for constructing pseudo-labeled data that can support more balanced capability growth of the solver.

### B.4 Skill Distribution of Constructed Pseudo-Labeled Data

Figure 7 shows the skill distribution statistics of the pseudo-labeled VQA data constructed by RISE every five training steps. Overall, the generated data do not collapse to a single dominant skill, which is consistent with our quantitative findings in the main paper. Instead, the skill distribution remains relatively balanced throughout training, while the emphasis on different skills varies across stages.

These variations are expected and meaningful. In each construction round, we generate pseudo-labeled data from a batch of 256 randomly sampled images, so the instantaneous skill distribution is partly influenced by the content of the sampled images. Nevertheless, because the image sampling is random, the persistent multi-skill coverage across different steps indicates that the observed distributions are not caused by fixed dataset bias, but instead reflect the effect of skill-aware dynamic balancing. In other words, RISE dynamically regulates question generation across different skill types, thereby avoiding the severe mode collapse observed in uncontrolled self-evolution.

## C Additional Experiments and Analysis

### C.1 Analysis of Supervisor Precision and Recall

We further analyze the supervisor during Qwen3-VL self-evolution, using Qwen3.5-Plus as an external reference. A sample is treated as problematic if its question is invalid or its pseudo-label is incorrect. Problematic samples are the positive class: rejecting one is a true positive, while accepting one is a false negative.

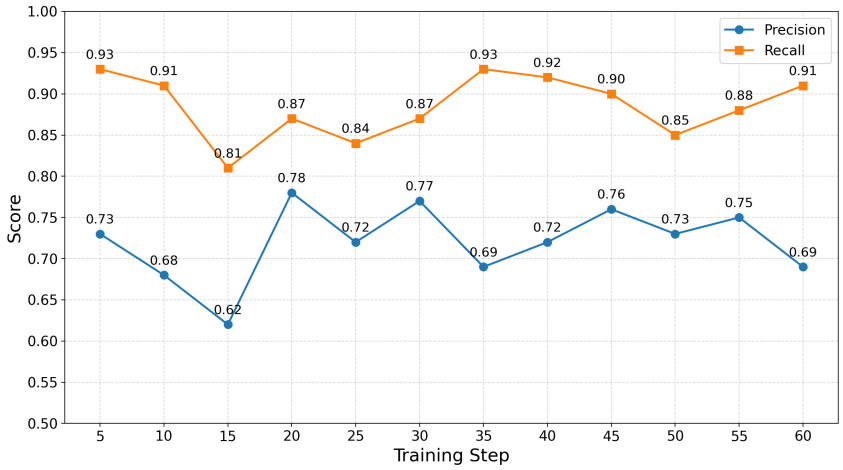


Figure 8: Precision and recall of the supervisor for detecting problematic samples during Qwen3-VL self-evolution. Problematic samples are defined as generated VQA samples that are not valid and correct according to Qwen3.5-Plus. The supervisor maintains consistently high recall across training, while precision is relatively lower but remains stable overall.

Figure 8 shows an average precision of 0.72 and recall of 0.885. Recall ranges from 0.81 to 0.93 across training, and the false-negative rate is 0.09 at Step 60. The average false-negative rate of 0.115 measures the fraction of problematic candidates that escape filtering; it is not the error rate among accepted samples. The lower precision indicates that the supervisor also rejects some valid samples.

The shared supervisor performs a different task from answer generation: it receives a candidate answer and checks whether it is supported by the image and question. The solver is rewarded for matching the majority-voted pseudo-label, with no explicit reward for approving its own answers. However, shared parameters can still lead to correlated errors. An incorrect answer may pass verification, enter training, and affect the solver used as the supervisor in the next cycle.

The external-reference results show no monotonic increase in the false-negative rate over the evaluated 60 steps. They support the usefulness of the filter, but do not rule out self-confirmation or establish that every accepted pseudo-label is correct.

### C.2 Why Can Self-Evolution Improve with Imperfect VQA Data?

Majority voting provides a consensus signal from the model's own response distribution. When the model already assigns higher probability to the correct answer than to competing answers, Maj@N can reduce sampling noise and provide a better target than a single sampled response. If the model instead consistently favors an incorrect answer, voting can reinforce that error. The method therefore depends on the base model's existing capability.

The ablations show both outcomes. On Qwen2.5-VL, removing the supervisor causes the valid-and-correct rate to fall below 10% in later stages, while AVG drops from 50.43 at Step 20 to 44.85 at Step 60, below the base score of 48.17. On the stronger Qwen3-VL, the corresponding quality rate remains around 30%, and the model without supervision still improves from 53.38 to 56.19. Full RISE further improves AVG to 57.37 (Appendix C.4). These results suggest that tolerance to noisy self-generated supervision depends on the base model, rather than a universal pseudo-label accuracy threshold.

Generated questions may also exercise useful visual reasoning skills even when some final answers are imperfect. However, the current experiments do not isolate this effect from the benefit of majority-voted targets. We therefore treat it as a possible explanation. In practice, quality control improves the usefulness of the training signal, while systematic errors in weak domains remain a limitation.

### C.3 Ablation on Role Alternation Granularity

We further study the effect of role alternation granularity in RISE. In our fine-grained role alternation, the hyperparameter $b$ denotes the number of update steps in each short alternation cycle. A larger $b$ corresponds to less frequent role switching, while a smaller $b$ makes the interaction between the questioner and the solver more online. In particular, $b = 20$ corresponds to the coarse-grained setting used in the main paper, $b = 5$ is our default fine-grained setting, and $b = 1$ represents an almost fully online-style alternation strategy.

To isolate the effect of alternation granularity, we keep the quality supervisor enabled for all variants but remove the skill-aware dynamic balancing module. This is because skill balancing is itself coupled with the alternation cycle: a smaller $b$ would update the recent skill statistics and apply balancing more frequently, making it difficult to attribute the performance change solely to role alternation granularity. Therefore, all variants in this study use the same quality-control mechanism but do not use skill balancing.

Table 3: Ablation study on the role alternation granularity $b$ using Qwen2.5-VL-7B-Instruct. All variants use the quality supervisor but remove skill-aware dynamic balancing. $b = 20$ denotes the coarse-grained setting, while smaller $b$ values correspond to more frequent role alternation. AVG denotes the sample-size weighted average over all benchmarks.

| Method | MMMU | MMVet | RealWorldQA | ChartQA | MathVerse | MathVision | MathVista | AVG |
|---|---|---|---|---|---|---|---|---|
| Base Model | 49.71 | 51.83 | 57.12 | 79.48 | 40.48 | 23.21 | 58.70 | 48.17 |
| $b = 1$, Step 20 | 52.28 | 59.17 | 57.91 | 81.20 | 43.73 | 27.47 | 58.80 | 50.94 |
| $b = 5$, Step 20 | 53.68 | 57.80 | 58.04 | 81.08 | 42.69 | 27.03 | 59.80 | 50.64 |
| $b = 10$, Step 20 | 52.51 | 57.34 | 58.30 | 80.88 | 44.21 | 25.44 | 58.90 | 50.59 |
| $b = 20$, Step 20 | 51.23 | 58.26 | 58.56 | 81.56 | 41.83 | 24.51 | 60.40 | 49.81 |
| $b = 1$, Step 40 | 51.23 | 55.96 | 58.95 | 82.08 | 43.15 | 27.40 | 57.70 | 50.76 |
| $b = 5$, Step 40 | 52.86 | 60.09 | 58.82 | 81.64 | 45.15 | 27.21 | 59.90 | 51.65 |
| $b = 10$, Step 40 | 54.73 | 52.75 | 59.08 | 81.76 | 44.26 | 27.18 | 57.50 | 51.19 |
| $b = 20$, Step 40 | 51.81 | 60.09 | 59.87 | 81.44 | 43.93 | 24.92 | 61.30 | 50.80 |
| $b = 1$, Step 60 | 52.98 | 54.59 | 60.00 | 81.76 | 42.44 | 26.95 | 59.90 | 50.71 |
| $b = 5$, Step 60 | 53.56 | 56.42 | 57.25 | 82.00 | 45.18 | 27.51 | 60.20 | 51.71 |
| $b = 10$, Step 60 | 53.79 | 52.29 | 58.30 | 81.32 | 44.21 | 27.47 | 59.30 | 51.17 |
| $b = 20$, Step 60 | 54.38 | 57.34 | 58.82 | 81.48 | 44.67 | 23.99 | 60.00 | 50.80 |

As shown in Table 3, role alternation granularity has a clear effect on the efficiency of self-evolution. When comparing the settings with $b = 5, 10, 20$, the performance consistently follows the trend of $b = 5 > b = 10 > b = 20$ at the same training step. Specifically, $b = 5$ achieves the highest AVG among these three settings at Step 20, Step 40, and Step 60, followed by $b = 10$, while the coarse-grained setting $b = 20$ performs the worst. This shows that more frequent role alternation consistently improves training efficiency by reducing the feedback lag between the questioner and the solver. This allows the questioner to adapt to a more up-to-date solver and enables the solver to learn from more timely generated data. At Step 20, the fully online-style setting with $b = 1$ further achieves the best AVG of 50.94, suggesting that extremely frequent interaction can accelerate early-stage adaptation.

However, the fully online-style setting does not continue to improve in later stages. The AVG of $b = 1$ remains almost unchanged from Step 20 to Step 60, changing from 50.94 to 50.76 and 50.71. This suggests that overly frequent alternation may make the interaction between the two roles too reactive. Since the questioner reward depends on the current solver's response consistency, updating the solver after every single step makes the capability boundary highly non-stationary. As a result, the questioner may overfit to transient weaknesses of the solver rather than constructing a stable and progressively informative curriculum. Meanwhile, the solver also has less opportunity to consolidate the pseudo-labeled data generated under the current questioner before the data distribution changes again.

In contrast, the moderate fine-grained setting $b = 5$ achieves the best long-term performance, reaching 51.65 at Step 40 and 51.71 at Step 60. This shows that $b = 5$ provides a better trade-off between feedback freshness and training stability. Compared with the coarse-grained setting $b = 20$, it substantially shortens the feedback loop; compared with the fully online setting $b = 1$, it still

preserves a short but stable phase for the questioner to adapt and for the solver to consolidate. These results support our choice of $b = 5$ as the default fine-grained alternation granularity in RISE.

### C.4 Additional Ablation Study on Qwen3-VL

We further conduct an ablation study on Qwen3-VL-8B-Instruct to examine whether the contribution of each component remains consistent on a stronger backbone. The dataset order follows the main paper, and we report the results at Step 60. Here, "w/o Skill" denotes removing the skill-aware dynamic balancing module, "w/o Sup." denotes removing the quality supervisor, and "w/o Fine-grained" denotes replacing fine-grained role alternation with coarse-grained alternation.

Table 4: Additional ablation study on Qwen3-VL-8B-Instruct at Step 60. AVG denotes the sample-size weighted average over all benchmarks.

| **Method** | **MMMU** | **MMVet** | **RealWQA** | **ChartQA** | **MathVerse** | **MathVision** | **MathVista** | **AVG** |
|---|---|---|---|---|---|---|---|---|
| RISE | 59.63 | 62.84 | 70.33 | 83.64 | 49.11 | 36.10 | 68.70 | 57.37 |
| RISE w/o Skill | 56.94 | 63.30 | 69.02 | 83.32 | 49.52 | 32.95 | 67.40 | 56.36 |
| RISE w/o Sup. | 56.71 | 62.84 | 68.89 | 82.92 | 49.47 | 32.65 | 67.80 | 56.19 |
| RISE w/o Fine-grained | 56.13 | 60.55 | 68.50 | 82.72 | 48.12 | 32.95 | 67.60 | 55.65 |

As shown in Table 4, removing any component leads to a lower overall weighted average, confirming that the three designs all contribute to the final performance on Qwen3-VL-8B-Instruct. Compared with the full RISE model, removing skill-aware dynamic balancing, the quality supervisor, and fine-grained role alternation decreases AVG by 1.01, 1.18, and 1.72 points, respectively.

It is worth noting that, unlike the Qwen2.5-VL ablation in the main paper, removing the supervisor on Qwen3-VL does not cause a severe late-stage collapse. We attribute this difference to the stronger base capability of Qwen3-VL. As shown in Fig. 3, although the quality of generated questions still decreases without the supervisor, Qwen3-VL can retain a non-trivial proportion of valid-and-correct samples in later stages, with the final ratio remaining around 30%. In contrast, Qwen2.5-VL drops to below 10% in the late stage, making the accumulated noisy supervision much more harmful. Therefore, the effect of the supervisor is closely related to the base model's ability to generate and filter reliable self-evolving data: stronger models are more robust to quality degradation, but the supervisor still provides clear gains by improving the reliability of pseudo-labeled supervision.

### C.5 Stability across Independent Runs

We conduct four additional complete training runs on Qwen3-VL-8B-Instruct with the same configuration, giving five independent runs in total. Table 5 reports the individual results at Step 60. AVG is $57.44 \pm 0.28$, and every run exceeds the base model score of 53.38. Performance varies across individual benchmarks, but the overall improvement is consistent.

Table 5: Results of five independent Qwen3-VL-8B-Instruct runs at Step 60. Mean and standard deviation are reported separately for readability. AVG is the sample-size weighted average.

| Model / Run | MMMU | MM-Vet | RealWQA | ChartQA | MathVerse | MathVision | MathVista | AVG |
|---|---|---|---|---|---|---|---|---|
| Original | 59.63 | 62.84 | 70.33 | 83.64 | 49.11 | 36.10 | 68.70 | 57.37 |
| Run 1 | 58.58 | 61.47 | 68.89 | 83.20 | 50.25 | 36.50 | 69.10 | 57.59 |
| Run 2 | 59.28 | 62.39 | 69.67 | 82.48 | 50.15 | 35.65 | 68.60 | 57.29 |
| Run 3 | 58.93 | 61.93 | 69.67 | 83.04 | 50.48 | 37.21 | 68.40 | 57.82 |
| Run 4 | 57.53 | 60.55 | 69.93 | 83.16 | 50.23 | 34.69 | 68.70 | 57.11 |
| Mean | 58.79 | 61.84 | 69.70 | 83.10 | 50.04 | 36.03 | 68.70 | 57.44 |
| Std. | 0.81 | 0.88 | 0.53 | 0.42 | 0.54 | 0.94 | 0.25 | 0.28 |

### C.6 Validation of the Evaluation Protocol

We compare Qwen3-Max with benchmark-specific evaluators on the same Qwen3-VL Base and RISE-Step60 outputs. Model responses and the deterministically extracted `\boxed{}` answers

are fixed, so only the scorer changes. We use task-specific deterministic scoring for MMMU, ChartQA, and MathVision, normalized exact match for RealWorldQA, and answer normalization followed by deterministic correctness rules for MathVista. These comprise benchmark-provided or benchmark-compatible evaluators, depending on the dataset.

Table 6 covers 7,823 samples. The two scoring protocols agree on 92.50% of Base predictions and 95.42% of RISE predictions, with agreement above 90% on each benchmark. RISE improves over Base on all five benchmarks under both protocols. The sample-weighted gain is +4.86 with Qwen3-Max and +7.16 with benchmark-specific scoring. Thus, the positive gain remains when the scorer changes, although its magnitude depends on the scoring rules.

Table 6: Controlled rescoring of fixed Qwen3-VL outputs. RISE denotes Step 60. Agreement is the fraction of samples assigned the same correctness label by the two scorers. The aggregate is weighted by sample count.

| | | Qwen3-Max | | | Benchmark-specific | | | Agreement | |
|---|---|---|---|---|---|---|---|---|---|
| Benchmark | $N$ | Base | RISE | $\Delta$ | Base | RISE | $\Delta$ | Base | RISE |
| MMMU | 857 | 55.31 | 59.63 | +4.32 | 50.64 | 58.11 | +7.47 | 94.40% | 98.72% |
| RealWorldQA | 765 | 67.45 | 70.33 | +2.88 | 66.54 | 68.76 | +2.22 | 98.69% | 99.08% |
| ChartQA | 2,500 | 81.44 | 83.64 | +2.20 | 83.36 | 86.60 | +3.24 | 90.48% | 92.28% |
| MathVision | 2,701 | 27.32 | 36.10 | +8.78 | 24.51 | 37.91 | +13.40 | 92.63% | 97.15% |
| MathVista | 1,000 | 65.80 | 68.70 | +2.90 | 59.00 | 62.60 | +3.60 | 90.80% | 93.00% |
| Weighted AVG | 7,823 | 56.52 | 61.38 | +4.86 | 54.70 | 61.86 | +7.16 | 92.50% | 95.42% |

MM-Vet and MathVerse use LLM-assisted evaluation protocols. We also report exact-match or quick-match checks in Table 7, but these are diagnostic scores rather than reproductions of the full official protocols. They are excluded from the agreement aggregate above.

Table 7: Diagnostic exact-match/quick-match checks on fixed Qwen3-VL Base and RISE-Step60 outputs. These scores do not reproduce the full official LLM-assisted evaluation protocols.

| | | Qwen3-Max | | | Diagnostic matching | | | Agreement | |
|---|---|---|---|---|---|---|---|---|---|
| Benchmark | $N$ | Base | RISE | $\Delta$ | Base | RISE | $\Delta$ | Base | RISE |
| MM-Vet | 218 | 60.55 | 62.84 | +2.29 | 25.23 | 30.28 | +5.05 | 63.30% | 71.10% |
| MathVerse | 3,940 | 46.75 | 49.11 | +2.36 | 45.23 | 53.43 | +8.20 | 87.66% | 92.79% |

### C.7 Longer Self-Evolving Training

We continue training the original Qwen3-VL Step60 checkpoint to Steps 80 and 100 under the same configuration. Table 8 shows that AVG increases from 57.37 to 57.90 and 58.05. The gains over the two additional 20-step intervals are +0.53 and +0.15, indicating diminishing returns.

The later gains are task-dependent. MathVerse improves from 49.11 at Step 60 to 51.24 at Step 100, while several other benchmarks plateau or fluctuate. These results show no aggregate collapse within 100 steps, but do not imply that longer training improves every capability or that gains continue indefinitely.

### C.8 Computational Cost

We compare RISE with our VisPlay reproduction using the same backbone, image set, update budget, hardware, and evaluation protocol. Both runs use 8 NVIDIA H20 GPUs with 96 GB memory. Table 9 reports end-to-end wall-clock time, including data generation and filtering. RISE takes 73 hours, compared with 65 hours for VisPlay, an increase of 12.3%. These correspond to 584 and 520 H20 GPU-hours, respectively.

The additional time comes from supervisor verification and more frequent role alternation. Under the same setting, AVG improves from 49.96 to 52.26. Sharing the supervisor with the solver avoids a separate verifier model, but does not remove the cost of verification inference.

Table 8: Longer self-evolution on Qwen3-VL-8B-Instruct. Steps 80 and 100 continue the original Step60 run with the same configuration.

| Model | MMMU | MM-Vet | RealWQA | ChartQA | MathVerse | MathVision | MathVista | AVG |
|---|---|---|---|---|---|---|---|---|
| Base Model | 55.31 | 60.55 | 67.45 | 81.44 | 46.75 | 27.32 | 65.80 | 53.38 |
| RISE-Step20 | 55.54 | 61.93 | 68.50 | 82.28 | 48.65 | 31.28 | 65.60 | 55.17 |
| RISE-Step40 | 58.11 | 62.39 | 70.85 | 82.24 | 49.06 | 32.80 | 67.70 | 56.15 |
| RISE-Step60 | 59.63 | 62.84 | 70.33 | 83.64 | 49.11 | 36.10 | 68.70 | 57.37 |
| RISE-Step80 | 58.93 | 61.93 | 70.20 | 83.36 | 51.07 | 35.95 | 69.20 | 57.90 |
| RISE-Step100 | 59.39 | 60.55 | 69.20 | 83.84 | 51.24 | 36.50 | 69.10 | 58.05 |

Table 9: End-to-end wall-clock cost under matched settings. All times are in hours on 8 H20 GPUs.

| Method | Questioner training and verification | Data generation and filtering | Solver training | Total |
|---|---|---|---|---|
| VisPlay (reproduced) | 36 | 16 | 13 | 65 |
| RISE | 40 | 20 | 13 | 73 |

### C.9 Reference Comparison with Prior Methods

Table 10: Comparison with prior self-evolving VLM methods using gain over the corresponding base model. † denotes our reproduced result, while other results are taken from the corresponding papers.

| **Method** | **MMMU** | **MMVet** | **RealWQA** | **ChartQA** | **MathVerse** | **MathVision** | **MathVista** |
|---|---|---|---|---|---|---|---|
| **Qwen2.5-VL-7B-Instruct** | | | | | | | |
| **RISE** | **+5.60** | **+5.97** | **+3.27** | +2.60 | **+4.47** | **+5.11** | +2.40 |
| Vision-Zero | +1.60 | – | +0.40 | +1.10 | +3.00 | +3.50 | **+3.90** |
| EvoLMM | +0.90 | – | – | **+2.70** | +1.10 | +0.90 | +2.06 |
| VisPlay† | +3.38 | +4.59 | +1.96 | +1.96 | +2.62 | -0.40 | +1.80 |
| **Qwen3-VL-8B-Instruct** | | | | | | | |
| **RISE** | **+4.32** | **+2.29** | **+2.88** | **+2.20** | +2.36 | **+8.78** | **+2.90** |
| MM-Zero | +2.50 | +0.90 | +0.79 | +2.00 | **+4.50** | +8.10 | -0.50 |

Table 10 provides a reference comparison of RISE with VisPlay (CVPR 2026), EvoLMM (CVPR 2026 Findings), Vision-Zero (ICLR 2026), and MM-Zero, reporting gains over each method's corresponding base model under its own experimental setting. Since base-model implementations, training data, update budgets, and evaluation protocols differ, these cross-paper gains do not establish a method ranking or demonstrate that RISE is superior to all prior methods. Our VisPlay reproduction provides a controlled comparison: its result corresponds to the Step 60 *Coarse-grained w/o Sup.* setting in Table 2, which uses coarse-grained questioner–solver alternation without the quality supervisor or skill-aware balancing. It shares the same Qwen2.5-VL-7B-Instruct backbone, Vision-47K image set, training-update budget, and evaluation protocol with RISE. The effects of our design choices are evaluated through this matched comparison and the component ablations in Table 2.

## D Limitations

First, self-evolution relies on model-generated pseudo-labels and cannot guarantee their correctness. Majority voting and supervisor filtering reduce noise, but the shared solver and supervisor may make correlated errors, allowing incorrect labels to enter subsequent training. Second, the extended 100-step run shows diminishing and task-dependent gains in later stages. Longer training does not consistently improve every benchmark, and continued improvement is not guaranteed. Finally, this work is problem-driven, focusing on delayed role interaction, supervision degradation, and skill diversity collapse. Our experiments assess whether the proposed mechanisms address these issues and improve self-evolution under controlled settings, rather than aiming to establish state-of-the-art performance across benchmarks.