SKILL-π: SKILL-POLICY CO-EVOLUTION FOR SELF-IMPROVING MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training with explicit reasoning trajectories is important for enhancing the reasoning capabilities of Multimodal Large Language Models (MLLMs). However, obtaining high-quality reasoning trajectories typically requires costly human annotation or supervision from stronger external models. To reduce this cost, self-improvement paradigms have emerged, enabling models to generate, verify, and learn from their own reasoning experience. Despite their effectiveness, existing self-improvement methods remain largely trajectory-centric, repeatedly consuming sample-specific reasoning trajectories while leaving persistently unresolved samples with little contribution to subsequent improvement. In this work, we introduce Skill-π, *a **skill-p**oli**c**y co-evolution framework that turns self-generated multimodal experience into reusable reasoning skills and continuously expands their capability coverage.* In Skill-π, a shared MLLM policy jointly performs skill selection, utilization, and evolution, with stage-specific credit assignment yielding learning signals according to the downstream consequences of each decision. The newly-recovered successful experience further improves the parametric policy and evolves the skill library, allowing reusable skills and the policy to improve iteratively across evolution rounds. Experiments show that Skill-π achieves up to $3\times$ higher evolutionary efficiency and $2.7\times$ greater data efficiency than existing self-improvement methods. Across diverse benchmarks spanning mathematical, chart/document, and general multimodal reasoning, Skill-π also achieves stronger performance against representative multimodal reasoning models.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have demonstrated rapidly growing capabilities in visual understanding (Ye et al., 2025; Ren et al., 2024; Ye et al., 2023) and multimodal reasoning (Zhang et al., 2023; Dong et al., 2025; Shao et al., 2024b). Beyond scaling model capacity and pre-training data, post-training with explicit reasoning trajectories has become an important approach for eliciting complex reasoning behaviors (Wei et al., 2026; Shao et al., 2024a; Guo et al., 2025a). However, obtaining high-quality reasoning supervision typically necessitates human-authored rationales or stronger external models (Wen et al., 2026), thereby making such supervision costly to scale (Yeo et al., 2025; Peng et al., 2025). Recent self-improvement methods provide a promising alternative which enables models to generate, verify, and learn from their own reasoning experience (Zelikman et al., 2022; Cheng et al., 2025; Singh et al., 2023). By repeatedly translating turning verified trajectories into supervision for iterative fine-tuning (Yuan et al., 2023; Zhong et al., 2026) or reinforcement learning (Thawakar et al., 2025; Liu et al., 2025; Li et al., 2026a), these methods establish a scalable pathway toward continuous self-improvement. Despite their effectiveness, existing methods remain predominantly trajectory-centric, repeatedly generating and consuming sample-specific reasoning traces across successive rounds of training. This raises a fundamental question in iterative self-improvement: *what knowledge from model's own experience should persist across rounds of evolution?*

Previous studies have explored the collection and selection of trajectories from successful attempts (Zelikman et al., 2022; Singh et al., 2023; Yuan et al., 2023), the rationalization and refinement of failed trials (Zelikman et al., 2022; Cheng et al., 2025), or partial reuse of trajectories for subsequent resampling (Zhong et al., 2026). Nevertheless, two fundamental limitations remain. (1) Reasoning trajectories are an overly instance-specific carrier of persistent experience. As shown in

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.

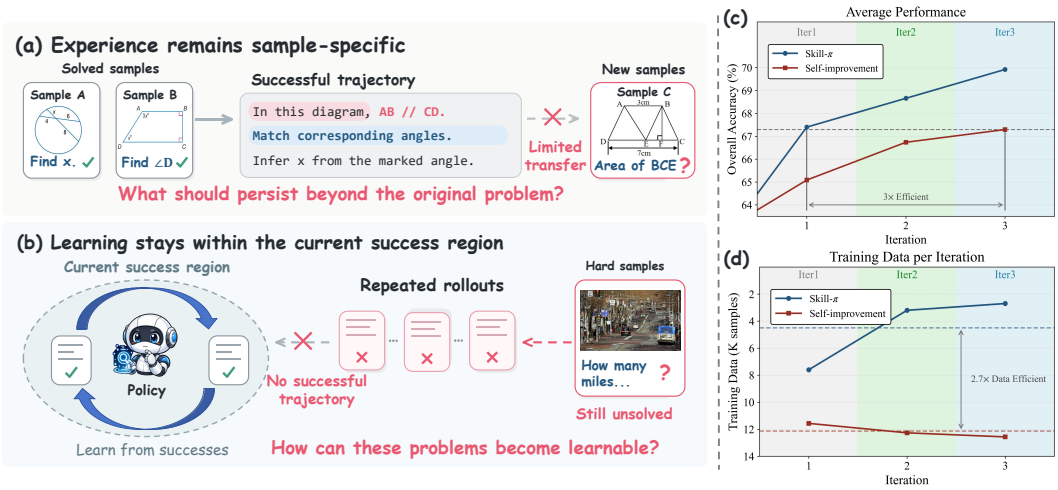


Figure 1: **Limitations of trajectory-centric self-improvement and the advantages of Skill-π.** (a) Reasoning trajectories remain tied to individual samples, limiting experience transfer. (b) Learning from successful trajectories confines improvement to the policy's current success region. (c) Skill-π achieves up to $3\times$ higher evolutionary efficiency than existing self-improvement methods. (d) Skill-π achieves $2.7\times$ higher data efficiency.

Fig. 1(a), sample-specific visual content and solution paths obscure the reusable reasoning patterns within trajectories, making them difficult to accumulate and transfer across samples and evolution rounds. (2) Trajectory-centric self-improvement remains largely confined to the model's current success region shown in Fig. 1 (b). Policy updates are primarily driven by trajectories that the model can already solve or locally recover, whereas persistently failing samples yield little actionable signal for directing subsequent exploration. Consequently, the model becomes increasingly proficient within its existing capability coverage without systematically expanding beyond it. This reveals a different view of self-improvement: successful experiences reveal capabilities to retain and reuse, while persistent failures expose where the model's capability coverage should expand.

Motivated by this perspective, we propose Skill-π, a skill-policy co-evolution framework for self-improving multimodal reasoning. Skill-π preserves self-generated experience at two complementary levels: successful trajectories are internalized into the parametric policy, while recurring reasoning behaviors across related successes are abstracted into a persistent library of transferable skills. These skills persist across rounds to guide subsequent exploration, while the newly-generated success experience further improves the policy and refines the skill library, forming a reciprocal cycle of skill and policy evolution. To optimize this co-evolutionary process, Skill-π trains a shared MLLM policy for skill selection, utilization, and evolution using GRPO (Shao et al., 2024c). Instead of supervising each stage independently, we assign stage-specific credit based on interaction outcomes, and jointly optimize skill decisions and policy learning via the consequences of generated experiences. Figs. 1(c) and (d) preview the resulting efficiency gains: Skill-π matches the three-iteration performance of previous self-improvement baselines in just one iteration, while these baselines consume $2.7\times$ more self-generated training data per iteration on average. In summary, our main contributions are as follows:

- We show that recurring successful behaviors in self-generated experience can be abstracted into transferable skills that persist across rounds and guide subsequent exploration. With this finding, we propose Skill-π, a skill-policy co-evolution framework that jointly accumulates experience in skill library and parametric policy.
- We introduce stage-specific credit assignment for co-evolving skills and policy. A shared MLLM policy performs skill selection, utilization, and evolution, with each stage being optimized according to its corresponding downstream interaction outcomes.
- Extensive experiments demonstrate that Skill-π achieves stronger multimodal reasoning performance, with up to $3\times$ higher evolutionary efficiency and $2.7\times$ greater data efficiency than existing self-improvement methods.

2 Related Work

2.1 Post-training for Multimodal Reasoning

Multimodal reasoning is strengthened through supervised fine-tuning on synthesized reasoning trajectories (Xu et al., 2025; Yao et al., 2026; Guo et al., 2025b) and reinforcement learning with verifiable rewards (Huang et al., 2026b; Meng et al., 2025; Chen et al., 2025). Cross-modal formalization transfers textual reasoning supervision to visual tasks (Yang et al., 2025), while reasoning-oriented initialization and reward-driven optimization develop longer reasoning processes (Huang et al., 2026b; Deng et al., 2026). Step-wise rewards further shape intermediate reasoning (Zhang et al., 2025). These approaches improve the supervision and optimization of reasoning behavior. Skill-π complements them with persistent skill memory that makes strategies distilled from prior experience explicitly available during the generation of new training trajectories.

2.2 Self-Improvement for Multimodal Reasoning

Self-improvement bootstraps reasoning by repeatedly sampling, filtering, and learning from model-generated solutions (Zelikman et al., 2022; Yuan et al., 2023; Singh et al., 2023; Liu et al., 2024). Beyond retaining successful attempts, answer-conditioned rationalization and reflection recover unsuccessful solutions (Zelikman et al., 2022; Cheng et al., 2025; Dou et al., 2024), while visual-attention-based selection and prefix reuse improve subsequent exploration (Zhong et al., 2026). Difficulty-aware data allocation, guided sampling, and head-tail rebalancing address the uneven coverage of hard problems in self-generated supervision (Tong et al., 2024; Ding et al., 2025; Guo et al., 2026). Together, these studies show that self-improvement depends on both trajectory quality and which problems contributing training experience. Skill-π addresses this challenge through cross-instance strategy transfer, where it consolidates related successes into reusable skills and uses them to steer exploration on problems that remain unsolved in repeated sampling.

2.3 Skill-Based Experience Reuse

Persistent experience stores empower agents to reuse executable procedures and natural-language insights across tasks (Wang et al., 2023; Zhao et al., 2024; Zhai et al., 2025; Zhang et al., 2024b). Recent work brings skill reuse into training by coupling library evolution with policy optimization and learning how to select, apply, and distill skills (Xia et al., 2026; Shi et al., 2026; Ma et al., 2026; Liu et al., 2026). Skills also serve as a training-time interface for task synthesis and verification, allowing the curriculum to co-evolve with the library (Huang et al., 2026a; Alzubi et al., 2026; Ouyang et al., 2026; Zhang et al., 2026b). Within this line of work, Skill-π focuses on visually grounded strategy transfer, where skills consolidate reasoning patterns across related successful instances, while visual-state compatibility guides their reuse to obtain new learning experience from previously unsuccessful multimodal exploration.

3 Method

We present Skill-π, a skill-policy co-evolution framework that augments a MLLM policy with a persistent library of reusable reasoning skills, as illustrated in Fig. 2. At each evolution round, the current skill library guides the policy to generate new reasoning experience, while successful experience is subsequently used to update both the parametric policy and the skill library. A shared MLLM conducts the learnable behaviors involved in this process, including skill selection, utilization, and evolution. We first introduce the problem setup and skill representation in Sec. 3.1, then describe how Skill-π constructs and reuses skills for experience generation in Sec. 3.2, and finally present the stage-specific optimization and iterative skill-policy co-evolution in Sec. 3.3.

3.1 Preliminaries

Given a multimodal training set $\mathcal{D} = \{(I_i, q_i, a_i)\}_{i=1}^{N}$, Skill-$\pi$ optimizes a MLLM policy $\pi_\theta^{(t)}$ and a persistent skill library $\mathcal{S}^{(t)} = \{s_j^{(t)}\}_{j=1}^{M_t}$ at evolution round t. Each skill is represented as $s_j = (v_j, \delta_j, \rho_j, u_j)$, where $v_j = (\text{domain}_j, \text{pattern}_j, \text{operation}_j)$ characterizes the visual state under

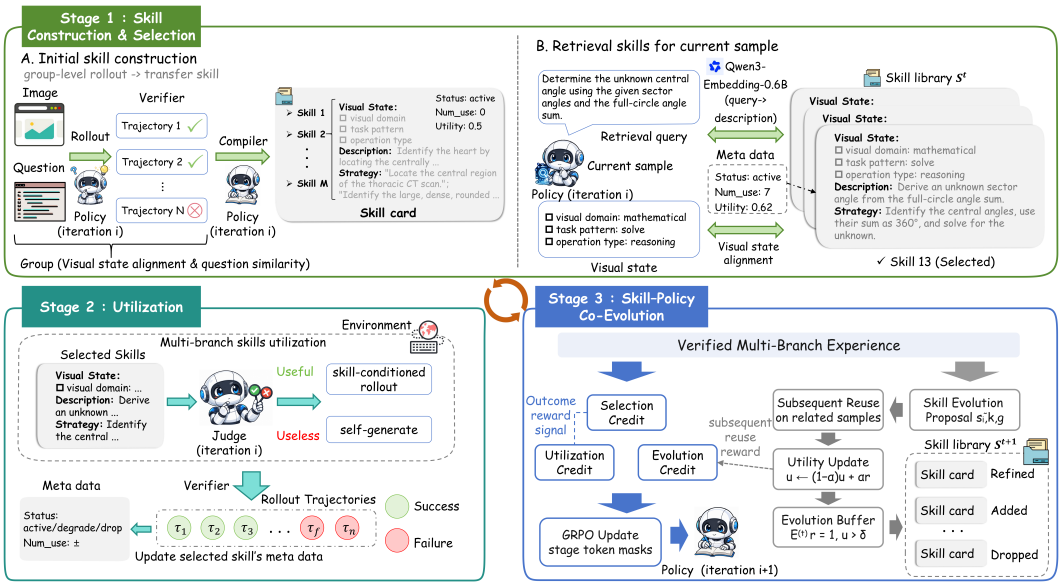


Figure 2: **Overview of Skill-π.** Skill-π constructs reusable skills from cross-sample experience, uses them to guide multi-branch reasoning, and applies stage-specific credit to skill selection, utilization, and evolution. The policy and skill library are iteratively updated through skill-policy co-evolution.

which the skill is applicable. The skill description δ_j specifies its reusable reasoning behavior and applicability, ρ_j describes the corresponding reasoning strategy, and u_j records its empirical utility from previous interactions. We initialize the skill library from verified successful experience $\mathcal{T}^{(0)}$, which is abstracted across related samples to construct initial skill library $\mathcal{S}^{(0)}$.

3.2 Skill-π Workflow

Skill-π accumulates reusable reasoning strategies from successful experience and transfers them to unresolved samples in subsequent exploration. Starting from the successful trajectory in Sec. 3.1, we construct a skill library through cross-sample abstraction and use it to guide policy interaction.

Skill Construction. For each successful sample, we derive a visual state $v_i = (\text{domain}_i, \text{pattern}_i, \text{operation}_i)$ from its image and question. Samples with the same visual state are grouped according to question-embedding similarity. For each group $\mathcal{G}_j$, recurring reasoning behaviors across successful trajectories are distilled into a skill description δ_j and a transferable reasoning strategy ρ_j, forming $s_j = (v_j, \delta_j, \rho_j, u_0)$, where u_0 is the initial empirical utility. This cross-sample abstraction reduces sample-specific solution leakage and preserves reasoning strategies that recur across related problems. The resulting skills form the initial library $\mathcal{S}^{(0)} = \{s_j\}_{j=1}^{M_0}$.

Skill Selection. Given a training sample $x_i = (I_i, q_i)$ with visual state v_i, we first filter the current skill library by visual-state compatibility. Specifically, skills satisfying $v_j = v_i$ form the candidate set $\mathcal{C}_i^{(t)} = \{s_j \in \mathcal{S}^{(t)} \mid v_j = v_i\}$. The shared policy then generates a skill-oriented retrieval query z_i, which is used to retrieve relevant skill descriptions:

$$\mathcal{B}_i^{(t)} = \text{Top}_{L, s_j \in \mathcal{C}_i^{(t)}} \text{Sim}\left(E(z_i), E(\delta_j)\right), \tag{1}$$

where $E(\cdot)$ denotes a frozen text encoder and L is the retrieval budget, distinct from the number K of selection decisions. Conditioned on the same retrieved candidate set $\mathcal{B}_i^{(t)}$, the policy selects distinct skills $\{s_{i,k}^*\}_{k=1}^{K}$. Each selected skill defines a distinct utilization branch, allowing alternative skill choices to be evaluated under a shared retrieval context. For samples that remain unresolved under unguided policy sampling, this retrieval process transfers strategies distilled from related successful samples to guide further exploration.

Skill Utilization. Each selected skill $s^*_{i,k}$ defines a skill-conditioned reasoning branch. Rather than generating a single trajectory for each selection, the policy sampling G reasoning trajectories $\{\tau_{i,k,g}\}_{g=1}^{G}$ derived from $\pi_\theta^{(t)}(\cdot \mid x_i, s^*_{i,k})$ under the same selected skill. The skill description $\delta^*_{i,k}$ and strategy $\rho^*_{i,k}$ are provided as optional guidance, while the policy may ignore guidance that conflicts with the visual evidence. Each trajectory is evaluated by the task verifier as $r_{i,k,g} = \mathcal{V}(\mathrm{Ans}(\tau_{i,k,g}), a_i)$. This associates each skill-selection decision with multiple downstream reasoning outcomes, which are later used to derive stage-aligned credit for skill selection and utilization. When skill-guided utilization succeeds on a previously unresolved sample, the resulting trajectory becomes new verified experience for subsequent policy and skill evolution.

Skill Evolution Proposal. After completing all K×G utilization rollouts, the shared policy generates one task-level proposal $\tilde{s}_i$ from $\mathcal{O}_i = \{(s^*_{i,k}, \tau_{i,k,g}, r_{i,k,g})\}_{k,g}$, which collects the selected skill, reasoning trajectory, and task reward for every skill branch and rollout. This proposal compares skill choices, successful strategies, and remaining failures to summarize reusable experience. Proposals from related tasks accumulate until they satisfy the unique-sample support and consistency requirements of skill construction. They are then jointly abstracted into a skill through the same construction procedure, with provenance links retained for subsequent credit assignment.

3.3 Skill-Policy Co-Evolution

Skill-π iteratively co-evolves the policy and skill library through self-generated experience. At evolution round t, the current library $\mathcal{S}^{(t)}$ guides the policy $\pi_\theta^{(t)}$ through skill selection, utilization, and evolution. The resulting task outcomes provide stage-specific learning signals for policy optimization, while successful proposals are consolidated to update the library from $\mathcal{S}^{(t)}$ to $\mathcal{S}^{(t+1)}$.

3.3.1 Stage-Specific Credit Assignment.

The multi-stage rollout structure induces different comparison groups for different policy decisions. For skill utilization, the selected skill $s^*_{i,k}$ is fixed, and the reasoning trajectories $\{\tau_{i,k,g}\}_{g=1}^{G}$ are compared within the same skill-conditioned context, isolating variation in executing skills. For skill selection, each selected skill defines a branch whose downstream value is aggregated over its utilization rollouts, and the resulting K branch-level outcomes are compared to assign credit to the selection decisions. This decomposition separates credit for *which skill to use* from credit for *how well the selected skill is utilized*.

Skill Selection Credit. The quality of a skill-selection decision is measured by the downstream performance of the corresponding utilization branch. For each selected skill $s^*_{i,k}$, we first aggregate its G task rewards as $\bar{r}_{i,k} = \frac{1}{G}\sum_{g=1}^{G} r_{i,k,g}$, and then compare the resulting branch-level outcomes across the K selection decisions:

$$A^{\mathrm{sel}}_{i,k} = \frac{\bar{r}_{i,k} - \mu^{\mathrm{sel}}_i}{\sigma^{\mathrm{sel}}_i + \eta}, \qquad \mu^{\mathrm{sel}}_i = \frac{1}{K}\sum_{k=1}^{K} \bar{r}_{i,k}, \tag{2}$$

where σ^{sel}_i denotes the standard deviation of K branch-level rewards. This advantage assigns credit to *which skill is selected* based on its expected downstream utility, while marginalizing the stochastic variation among individual utilization rollouts.

Skill Utilization Credit. For each selected skill $s^*_{i,k}$, we treat its G utilization trajectories generated under this skill as an independent comparison group. Let $r_{i,k,g}$ denote the task reward of trajectory $\tau_{i,k,g}$. We compute the utilization advantage by normalizing rewards within the same skill-conditioned branch:

$$A^{\mathrm{util}}_{i,k,g} = \frac{r_{i,k,g} - \mu_{i,k}}{\sigma_{i,k} + \eta}, \qquad \mu_{i,k} = \frac{1}{G}\sum_{g=1}^{G} r_{i,k,g}, \tag{3}$$

where $\sigma_{i,k}$ denotes the corresponding reward standard deviation and $\eta > 0$ is a numerical stabilizer, distinct from the GRPO clipping radius ϵ. Since all trajectories in the group share the same sample x_i and selected skill $s^*_{i,k}$, the resulting advantage attributes relative performance to variation in skill utilization rather than skill selection.

Skill Evolution Credit. A task-level proposal receives credit through the skill constructed from its group. Let $j(i)$ identify this skill and $\hat{u}_i = u_{j(i)}$ its utility, which is updated once per subsequent skill-query interaction by $u_j \leftarrow (1-\alpha)u_j + \alpha\bar{r}_j$, where $\bar{r}_j$ averages its G rollout rewards. For an update batch $\mathcal{H}$ of feedback-bearing proposals from different construction groups, we compute

$$A_i^{\text{evo}} = \frac{\hat{u}_i - \mu_{\mathcal{H}}^{\text{evo}}}{\sigma_{\mathcal{H}}^{\text{evo}} + \eta}, \qquad \mu_{\mathcal{H}}^{\text{evo}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \hat{u}_i. \tag{4}$$

Here $\sigma_{\mathcal{H}}^{\text{evo}}$ is the corresponding standard deviation. All proposals contributing to a skill share its utility, while proposals awaiting construction or reuse feedback are excluded from this update.

3.3.2 Joint Skill-Policy Evolution.

Policy Evolution. The shared MLLM is optimized with the stage-specific advantages derived above. Let $\mathcal{M}^c$ denote the token mask corresponding to stage $c \in \{\text{sel}, \text{util}, \text{evo}\}$, such that each credit signal is applied only to tokens generated at its corresponding decision stage. For a stage-specific comparison group $\mathcal{G}^c$, we define the masked GRPO objective as

$$\mathcal{J}_{\text{GRPO}}^c(\theta) = \mathbb{E}\left[\frac{1}{|\mathcal{G}^c|}\sum_{b\in\mathcal{G}^c}\frac{1}{|\mathcal{M}_b^c|}\sum_{m\in\mathcal{M}_b^c}\min\left(\rho_{b,m}A_b^c, \text{clip}(\rho_{b,m}, 1-\epsilon, 1+\epsilon)A_b^c\right)\right], \tag{5}$$

where $\rho_{b,m} = \pi_\theta(y_{b,m} \mid h_{b,m})/\pi_{\theta_{\text{old}}}(y_{b,m} \mid h_{b,m})$, and b indexes decisions in the corresponding group. For proposals, the denominator uses their recorded generating policy. We obtain $\mathcal{J}_{\text{sel}}$, $\mathcal{J}_{\text{util}}$, and $\mathcal{J}_{\text{evo}}$ by applying Eq. 5 to their respective stage-specific groups and advantages. The overall policy objective combines the three stage-specific objectives:

$$\mathcal{J}(\theta) = \mathcal{J}_{\text{util}} + \lambda_{\text{sel}}\mathcal{J}_{\text{sel}} + \lambda_{\text{evo}}\mathcal{J}_{\text{evo}} - \beta D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right), \tag{6}$$

where λ_{sel} and λ_{evo} control the contributions of skill selection and skill evolution, respectively, and β controls the KL regularization toward the reference policy.

Skill Library Evolution. Task-level proposals with successful supporting experience enter an accumulation buffer:

$$\mathcal{E}^{(t)} = \{(x_i, \mathcal{O}_i, \tilde{s}_i) \mid \max_{k,g} r_{i,k,g} = 1\}. \tag{7}$$

Proposals are grouped by visual state and question similarity. Once a group contains at least m unique samples and supporting a consistent strategy, the skill-construction pipeline abstracts it into a skill and incorporates it into the library. Groups below this support threshold remain buffered. Subsequent reuse updates the constructed skill's utility and supplies credit to its source proposals, enabling skill construction to precede utility-based optimization.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate Skill-π on ten multimodal reasoning benchmarks. The controlled self-improvement experiments use SLAKE (Liu et al., 2021), VQA-RAD (Lau et al., 2018), and Geometry3K (Lu et al., 2021); the broader comparison uses MathVerse (Zhang et al., 2024a), MATH-Vision (Wang et al., 2024), MathVista (Lu et al., 2024), We-Math (Qiao et al., 2025), Hallusion-Bench (Guan et al., 2024), ChartQA (Masry et al., 2022), and MMMU (Yue et al., 2024).

Baselines. We compare Skill-π with four groups of baselines. (1) Self-improvement methods include STaR (Zelikman et al., 2022), ReST$^{\text{EM}}$ (Singh et al., 2023), RFT (Yuan et al., 2023), R3V (Cheng et al., 2025), and VISTA (Zhong et al., 2026). (2) Closed-source MLLMs include GPT-4o (Achiam et al., 2023), OpenAI-o1 (Jaech et al., 2024), and Claude 3.7 Sonnet (Anthropic, 2025). (3) Open-source general MLLMs include the Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), InternVL2.5 (Chen et al., 2024), and InternVL3 (Zhu et al., 2025) families. (4) Open-source reasoning MLLMs include R1-VL-7B (Zhang et al., 2025), Vision-R1-7B (Huang et al., 2026b), R1-OneVision-7B (Yang et al., 2025), OpenVLThinker-7B (Deng et al., 2026), VLAA-Thinker-7B (Chen et al., 2025), MM-Eureka-Qwen-7B (Meng et al., 2025), Thyme-VL-7B (Zhang et al., 2026a), and ThinkLite-VL-7B (Wang et al., 2026).

Table 1: Performance of Qwen2.5-VL models with different self-improvement methods on medical and mathematical benchmarks. SFT-Seed denotes fine-tuning on self-generated seed trajectories, while SFT-Oracle uses trajectories distilled from Qwen3-VL-32B-Instruct as an approximate SFT upper bound. "Train Avg." reports the average number of training samples, and "Overall" denote the average accuracy, with subscripts indicating gains over SFT-Seed. Best results are in **bold**.

Methods	Train Avg.	Qwen2.5-VL-3B-Instruct				Qwen2.5-VL-7B-Instruct			
		SLAKE	VQA-RAD	Geo3K	Overall (Δ)	SLAKE	VQA-RAD	Geo3K	Overall (Δ)
SFT-Seed	1.0K	67.04	64.14	25.46	52.21	79.15	70.52	36.94	62.20
SFT-Oracle	1.3K	81.97	68.13	37.94	62.68↑10.47	82.25	70.12	41.43	64.60↑2.40
RFT	25.5K	79.15	70.12	28.79	59.35↑7.14	82.82	74.90	40.27	66.00↑3.80
STaR									
Iteration 1	0.9K	73.80	66.13	32.11	57.35↑5.14	80.00	71.31	37.44	62.92↑0.72
Iteration 2	1.0K	76.90	65.34	31.95	58.06↑5.85	80.00	72.11	37.94	63.35↑1.15
Iteration 3	1.1K	80.28	69.32	30.78	60.13↑7.92	84.23	69.72	37.10	63.68↑1.48
ReST^{EM}									
Iteration 1	8.5K	76.06	69.72	29.95	58.58↑6.37	82.82	73.31	35.77	63.97↑1.77
Iteration 2	9.9K	79.38	71.31	31.45	60.71↑8.50	84.79	74.49	39.77	66.35↑4.15
Iteration 3	10.7K	81.69	73.71	32.28	62.56↑10.35	84.51	75.30	38.94	66.25↑4.05
R3V									
Iteration 1	18.5K	78.21	65.74	29.62	57.86↑5.65	82.82	75.30	36.44	64.85↑2.65
Iteration 2	18.8K	82.82	74.10	32.61	63.18↑10.97	86.48	75.30	38.44	66.74↑4.54
Iteration 3	18.7K	81.41	69.32	32.78	61.17↑8.96	87.32	75.70	37.27	66.76↑4.56
VISTA									
Iteration 1	7.7K	80.85	74.10	32.28	62.41↑10.20	83.66	74.90	40.77	66.44↑4.24
Iteration 2	8.1K	83.38	72.51	34.28	63.39↑11.18	84.79	75.30	41.26	67.12↑4.92
Iteration 3	8.3K	84.23	76.10	37.27	65.87↑13.66	87.89	77.29	41.43	68.87↑6.67
Skill-π-SFT (Ours)									
Iteration 1	7.6K	82.54	73.71	35.94	64.06↑11.85	84.79	76.49	40.93	67.40↑5.20
Iteration 2	3.2K	84.23	75.30	37.60	65.71↑13.50	86.20	77.69	42.10	68.66↑6.46
Iteration 3	2.7K	**85.92**	**77.29**	**38.77**	**67.33↑15.12**	**88.73**	**78.09**	**42.93**	**69.92↑7.72**

Training Details. For a fair comparison with self-improvement methods, we use an SFT-only variant, Skill-π-SFT, without subsequent RL optimization. We follow the VISTA protocol (Zhong et al., 2026), using the same Qwen2.5-VL-3B/7B backbones and data splits, with 10 rollouts per query, a sampling temperature of 1.0, a maximum generation length of 2,048, and 3 self-improvement rounds. All models are trained for 3 epochs per round. For the full Skill-π, we use Vision-SR1-Cold-9K for cold-start initialization and Vision-SR1-47K for RL post-training (Li et al., 2026b). The cold-start model is fully fine-tuned for 3 epochs with a peak learning rate of 1×10^{-5}, while RL post-training adopts LoRA with rank 64, a scaling factor of 128, and a learning rate of 1×10^{-6}. For skill utilization, we retrieve five candidates, select two distinct skills, and sample eight responses per branch. All experiments are conducted with bfloat16 precision on 8 NVIDIA A800 GPUs. Additional implementation details are provided in Appendix A.

4.2 Main Results

Comparison with Self-Improvement Methods. Table 1 compares Skill-π with representative self-improvement training methods under the same Qwen2.5-VL-3B/7B backbones and training protocol. For a fair comparison, we report Skill-π-SFT, an SFT-only variant of our framework without subsequent RL optimization. Despite this simplified setting, Skill-π-SFT consistently outperforms STaR, ReST^{EM}, RFT, R3V, and VISTA across both model scales. After 3 self-improvement rounds, it improves over SFT-Seed by +15.12% and +7.72% on the 3B and 7B models, respectively, exceeding the strongest competing method VISTA, while using 67.5% fewer training samples than VISTA in the final round. Moreover, the amount of training data required by Skill-π-SFT progressively decreases across rounds, whereas its performance continues to improve. These results indicate that accumulating self-generated experience into reusable skills enables more effective and data-efficient policy improvement than repeatedly training on trajectory-level experience alone.

Comparison with General and Reasoning MLLMs. Table 2 shows that Skill-π improves both backbones across seven benchmarks. Skill-π-7B improves the average score over Qwen2.5-VL-7B by 7.2% and outperforms ThinkLite-VL-7B by 2.6%. The gains over the base model are largest on We-Math (+10.7%) and MATH-Vision (+9.4%), while extending to HallusionBench (+6.7%),

Table 2: Comparison with open-source and closed-source MLLMs across multimodal reasoning benchmarks. Skill-π-7B/8B are based on Qwen2.5-VL-7B/Qwen3-VL-8B, respectively.

Model	MathVerse	MathVision	MathVista	WeMath	HallBench	ChartQA	MMMU	Avg.
Closed-Source MLLMs								
GPT-4o	50.8	30.4	63.8	68.8	55.0	85.7	69.1	60.5
OpenAI-o1	57.0	60.3	73.9	–	–	83.1	77.6	–
Claude-3.7-Sonnet	52.0	41.3	66.8	72.6	55.4	56.5	75.0	59.9
Open-Source General MLLMs								
InternVL-2.5-8B	39.5	19.7	64.4	53.5	61.7	79.1	62.7	54.4
InternVL-3-8B	39.8	29.3	71.6	58.1	64.3	85.9	60.7	58.5
Qwen2.5-VL-7B	46.3	25.1	67.8	62.1	65.0	83.5	58.6	58.3
Qwen2.5-VL-7B-TIR	47.2	26.3	68.1	63.7	67.2	84.1	59.6	59.5
Qwen3-VL-8B	62.1	53.9	77.2	72.5	72.1	84.6	69.6	70.3
Qwen3-VL-8B-TIR	63.1	54.7	79.4	73.1	72.8	85.4	70.9	71.3
Open-Source Reasoning MLLMs								
R1-VL-7B	40.4	24.7	63.5	60.1	54.7	76.1	52.3	53.1
Vision-R1-7B	51.9	30.7	73.5	73.9	68.8	79.8	50.5	61.3
R1-OneVision-7B	46.4	29.9	64.1	61.8	67.5	77.8	52.9	57.2
OpenVLThinker-7B	45.7	26.3	71.2	66.7	70.2	78.4	58.4	59.6
VLAA-Thinker-7B	52.7	29.2	69.7	70.2	68.2	80.1	54.7	60.7
MM-Eureka-Qwen-7B	50.5	27.9	73.6	67.4	66.9	82.1	52.7	60.2
ThinkLite-VL-7B	52.1	32.9	75.1	69.3	70.9	84.8	55.5	62.9
Thyme-VL-7B	51.3	27.6	70.0	39.3	71.0	86.1	54.7	57.1
Skill-π-7B (Ours)	54.2	34.5	76.3	72.8	71.7	87.8	61.4	65.5
Skill-π-8B (Ours)	66.4	57.8	84.7	80.3	74.2	89.3	75.1	75.4

Table 3: Performance of Skill-π across self-improvement iterations based on Qwen2.5-VL-7B.

Model	MathVerse	MathVision	MathVista	WeMath	HallBench	ChartQA	MMMU	Avg.
Base model	46.3	25.1	67.8	62.1	65.0	83.5	58.6	58.3
Cold-start (SFT)	48.8	27.3	70.2	64.6	66.2	84.3	59.0	60.1
Iteration 1	50.7	29.8	72.3	67.5	68.1	85.5	59.9	62.0
Iteration 2	52.5	32.2	74.3	70.2	69.9	86.7	60.7	63.8
Iteration 3	54.2	34.5	76.3	72.8	71.7	87.8	61.4	65.5

ChartQA (+4.3%), and MMMU (+2.8%). With the stronger Qwen3-VL-8B backbone, Skill-π improves over the base and its TIR variant by 5.1% and 4.1%, respectively, and leads all listed open-source models on every benchmark. These results support the applicability of skill-based improvement across model scales and task domains.

Iterative Self-Improvement. Table 3 shows successive gains of 1.9%, 1.8%, and 1.7% in the average score over three iterations. Every benchmark improves at each round, yielding a cumulative gain of 5.4% over the cold-start model. The sustained gains indicate that the framework continues to benefit from successive rounds of skill-policy co-evolution, rather than obtaining its improvement only during initialization.

4.3 Analysis

Effective skill reuse requires selection, utilization, and evolution. As shown in Table 4, removing skill selection, utilization, or evolution reduces the average score by 2.6%, 3.2%, and 2.8%, respectively. Every removal consistently degrades performance across all benchmarks, and the largest average drop is removing utilization, underscoring the importance of learning to apply retrieved knowledge. The selection and evolution ablations further show that effective reuse depends on selecting relevant skills and updating the library as training proceeds.

Stage-specific credit improves learning beyond answer rewards. As shown in Table 4, seeping the full workflow but disabling selection or evolution credit reduces the average score by 1.1% or 1.7%. Disabling both increases the loss to 2.3%, showing that each signal contributes even when the other remains active. Therefore, answer-based optimization therefore does not fully replace direct feedback on skill selection and evolution, despite preserving both stages in the workflow.

Table 4: Ablation study of Skill-π on multimodal reasoning benchmarks. We evaluate the contributions of skill selection, skill utilization, skill evolution, and stage-specific credit assignment.

Setting	MathVerse	MathVision	MathVista	WeMath	HallBench	ChartQA	MMMU	Avg.
Skill-π	54.2	34.5	76.3	72.8	71.7	87.8	61.4	65.5
w/o Skill Selection	51.8	30.9	73.4	69.0	69.0	86.2	60.3	62.9
w/o Skill Utilization	51.0	30.3	72.6	68.0	68.6	85.7	60.0	62.3
w/o Skill Evolution	51.4	30.9	73.2	68.4	69.0	86.0	60.1	62.7
w/ $\lambda_{\text{sel}} = 0$	53.1	33.1	75.0	71.1	70.5	87.1	61.0	64.4
w/ $\lambda_{\text{evo}} = 0$	52.5	32.2	74.5	70.2	70.1	86.7	60.6	63.8
w/ $\lambda_{\text{sel}} = \lambda_{\text{evo}} = 0$	52.0	31.5	73.6	69.5	69.4	86.2	60.4	63.2
Cold-start (SFT)	48.8	27.3	70.2	64.6	66.2	84.3	59.0	60.1
Base Model	46.3	25.1	67.8	62.1	65.0	83.5	58.6	58.3

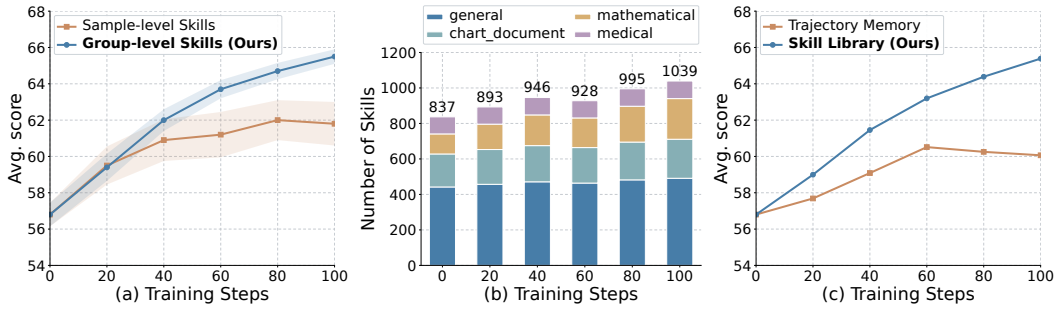


Figure 3: Skill abstraction and library dynamics during training. (a) Group-level versus sample-level skill construction. (b) Skill counts across four domains; labels indicate library totals. (c) Skill-library versus trajectory-memory performance. Shaded regions indicate standard deviation.

Shared skill abstraction sustains gains as isolated experiences plateau. Group-level skills continue improving as sample-level skills plateau, yielding an approximately 3.7% final advantage in Fig. 3(a). Likewise, the skill library sustains gains while trajectory memory levels off, reaching an approximately 5.3% final advantage in Fig. 3(c). Both comparisons favor consolidating shared reasoning patterns over retaining isolated experiences.

Continued improvement does not require monotonic library growth. Fig. 3(b) shows that the library grows from 837 to 1,039 skills across four domains, with a temporary decrease from 946 to 928. Performance continues to improve over this interval in Figs. 3(a) and 3(c), so a larger library is not necessary for every gain. This suggests that Skill-π benefits from selective refinement and consolidation rather than indiscriminate library expansion.

5 Conclusion

We introduced Skill-π, a skill-policy co-evolution framework that preserves self-generated multimodal experience as transferable reasoning skills. Its central principle is that successful experience should guide future exploration through reusable strategies, and also improve the parametric policy. By abstracting related successes into a persistent library, Skill-π transfers strategies to previously unresolved samples and incorporates newly recovered experience into subsequent evolution. Stage-specific credit trains a shared policy to select, utilize, and evolve skills according to their downstream consequences. Controlled SFT experiments show improved evolutionary and data efficiency, while the full framework improves both backbones across seven multimodal reasoning benchmarks. Ablations support the contributions of skill selection, utilization, and evolution, and show that answer rewards alone do not recover the benefits of stage-specific credit. Together, these results support reusable skill abstraction as a means of sustaining multimodal self-improvement beyond repeated learning from isolated trajectories. The present evaluation focuses on tasks with verifiable answers and a finite number of evolution rounds, while open-ended feedback and longer-term skill reuse remain future directions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, sentence restructuring, and improving the clarity and conciseness of the manuscript. We also used generative AI tools for code review, including checking implementation logic, identifying potential bugs, and suggesting improvements to code readability and organization. All AI-assisted code changes were manually reviewed and tested by the authors before use.

REPRODUCIBILITY STATEMENT

We document the data subsets, training procedures, evaluation protocols, and software frameworks in Appendix A, with numerical configurations in Table 5. All training and evaluation data used in this work are drawn from publicly available datasets. Skill construction, retrieval and utilization, and library evolution are detailed in Appendices C, C.3, and C.4, including prompts and update rules. Algorithm 1 summarizes the training procedure, and Appendix D.1 provides the derivation of stage-specific credit assignment. Hardware requirements and computational costs are reported in Appendix B.4. We will release the complete codebase and all supporting implementation files required to reproduce our experiments after acceptance.

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.

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*, 2026.

AI Anthropic. Claude 3.7 sonnet system card, 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.

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.

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

Kanzhi Cheng, Li YanTao, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8876–8892, 2025.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: Complex vision-language reasoning via iterative sft-rl cycles. *Advances in Neural Information Processing Systems*, 38:123817–123846, 2026.

Yiwen Ding, Zhiheng Xi, Wei He, Zhuoyuan Li, Yitao Zhai, Xiaowei Shi, Xunliang Cai, Tao Gui, Qi Zhang, and Xuan-Jing Huang. Mitigating tail narrowing in llm self-improvement via socratic-guided sampling. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10627–10646, 2025.

Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9062–9072. IEEE, 2025.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15394–15411, 2024.

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

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

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13869–13920, 2025b.

Xin Guo, Zhiheng Xi, Yiwen Ding, Yitao Zhai, Xiaowei Shi, Xunliang Cai, Tao Gui, Qi Zhang, and Xuan-Jing Huang. Counteracting the matthew effect in self-improvement of lvlms through head-tail re-balancing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22104–22121, 2026.

Siyuan Huang, Pengyu Cheng, Haotian Liu, Tao Chen, Yihao Liu, Jingwei Ni, Shijie Zhou, Ziyi Yang, Gangwei Jiang, Mengyu Zhou, et al. Skill self-play: Pushing the frontier of llm capability with co-evolving skills. *arXiv preprint arXiv:2607.22529*, 2026a.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026b.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):180251, 2018.

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*, 2026a.

Zongxia Li, Wenhao Yu, Chengsong Huang, Zhenwen Liang, Rui Liu, Fuxiao Liu, Jingxi Chen, Dian Yu, Jordan Boyd-Graber, Haitao Mi, et al. Vision-sr1: Self-rewarding vision-language model via reasoning decomposition and multi-reward policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 81753–81772, 2026b.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th international symposium on biomedical imaging (ISBI)*, pp. 1650–1654. IEEE, 2021.

Jiaqi Liu, Kaiwen Xiong, Peng Xia, Yiyang Zhou, Haonian Ji, Lu Feng, Siwei Han, Mingyu Ding, and Huaxiu Yao. Agent0-vl: Exploring self-evolving agent for tool-integrated vision-language reasoning. *arXiv preprint arXiv:2511.19900*, 2025.

Wei Liu, Junlong Li, Xiwen Zhang, Fan Zhou, Yu Cheng, and Junxian He. Diving into self-evolving training for multimodal reasoning. *arXiv preprint arXiv:2412.17451*, 2024.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, et al. Skillrevise: Improving llm-authored agent skills via trace-conditioned skill revision. *arXiv preprint arXiv:2606.01139*, 2026.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786, 2021.

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. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

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, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. 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.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, et al. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*, 2026.

Xiangyu Peng, Congying Xia, Xinyi Yang, Caiming Xiong, Chien-Sheng Wu, and Chen Xing. Regenesis: Llms can grow into reasoning generalists via self-improvement. In *International Conference on Learning Representations*, volume 2025, pp. 62484–62519, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070, 2025.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323. IEEE, 2024.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024b.

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

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026.

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.

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.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Advances in Neural Information Processing Systems*, 37:7821–7846, 2024.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

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.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *Advances in Neural Information Processing Systems*, 38: 118818–118850, 2026.

Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. First sft, second rl, third upt: Continual improving multi-modal llm reasoning via unsupervised post-training. *Advances in Neural Information Processing Systems*, 38:62293–62318, 2026.

Zichen Wen, Shaobo Wang, Yufa Zhou, Junyuan Zhang, Qintong Zhang, Yifeng Gao, Zhaorun Chen, Bin Wang, Weijia Li, Conghui He, et al. Efficient multi-modal large language models via progressive consistency distillation. *Advances in Neural Information Processing Systems*, 38: 69726–69753, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2376–2385. IEEE, 2025.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *Advances in Neural Information Processing Systems*, 38:29918–29952, 2026.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2841–2858, 2023.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 98891–98913, 2025.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

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.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, et al. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1859–1869. IEEE, 2025.

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, 2024a.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5348–5375, 2024b.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. In *International Conference on Learning Representations*, 2026a.

Zhiwei Zhang, Yudi Lin, Nikki Lijing Kuang, Linlin Wu, Xiaomin Li, Songtao Liu, and Fenglong Ma. Co-evolving skill generation and policy optimization. *arXiv preprint arXiv:2606.08755*, 2026b.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Qihuang Zhong, Liang Ding, Wenjie Xuan, Juhua Liu, Bo Du, and Dacheng Tao. Learn to think: Improving multimodal reasoning through vision-aware self-improvement training. *arXiv preprint arXiv:2605.11931*, 2026.

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.

APPENDIX CONTENTS

A Implementation Details **16**

B Additional Analysis **19**

B.1 Analysis of Credit Assignment 19

B.2 Analysis of Skill Retrieval and Selection 19

B.3 Analysis of Capability Expansion and Skill Utility 20

B.4 Computation Resource Analysis 21

B.5 Hyperparameter Sensitivity 21

C Skill Construction Details **22**

C.1 Visual State. 22

C.2 Group-Level Skill Construction. 23

C.3 Skill Selection and Utilization Details 24

C.4 Skill Evolution Details 26

D Algorithm Details **27**

D.1 From GRPO to Stage-Specific Credit Assignment 27

E Case Studies **30**

E.1 Case 1: Medical anatomy 30

E.2 Case 2: Geometry 34

E.3 Case 3: Table reasoning 38

F Limitations and Future Work **41**

A Implementation Details

Data Details. For the controlled comparison with self-improvement methods in Table 1, we follow the same data processing and evaluation protocol as VISTA (Zhong et al., 2026). The three datasets and their corresponding subsets are described below.

- **SLAKE** (Liu et al., 2021) is a bilingual medical visual question answering (VQA) benchmark containing both open-ended and closed-ended questions across multiple imaging modalities and anatomical regions. It requires models to interpret medical images in conjunction with the question. Following VISTA, we retain only the closed-ended English questions and their corresponding answers, resulting in 1,681 training samples and 355 test samples.
- **VQA-RAD** (Lau et al., 2018) contains 2,248 question–answer pairs associated with 315 radiology images. Its questions include open-ended responses and binary yes/no answers, requiring image interpretation in a clinical context. We use the same closed-ended subset as VISTA for both training and evaluation, with 940 training samples and 251 test samples.
- **Geometry3K** (Lu et al., 2021) contains 3,002 geometry problems that combine textual statements with geometric diagrams. Solving these problems requires relating the quantities and spatial relationships in the diagram to the mathematical constraints in the question. Following VISTA, we use the open-ended variant[1], in which the original multiple-choice problems are converted into questions requiring a directly generated answer. The training and test sets contain 2,101 and 601 samples, respectively.

No additional data filtering or re-splitting is introduced in this controlled setting.

For the broader multimodal reasoning evaluation in Table 2, we use the following seven benchmarks, covering visual mathematics, hallucination, chart reasoning, and multidisciplinary understanding. We follow their official evaluation splits unless otherwise specified.

- **MathVerse** (Zhang et al., 2024a) assesses whether models use diagram evidence when solving visual mathematics problems. It contains 2,612 source problems covering plane geometry, solid geometry, and functions, with each problem adapted into six versions that vary the information supplied through text and images. These versions range from text-only questions to problems whose essential information is available only in the diagram, enabling analysis of the dependence on visual evidence.
- **MATH-Vision** (Wang et al., 2024) contains 3,040 visually grounded mathematical problems collected from mathematics competitions. The benchmark spans 16 mathematical categories and five difficulty levels. It evaluates the ability to combine visual interpretation with mathematical reasoning across problems of increasing complexity.
- **MathVista** (Lu et al., 2024) contains 6,141 examples drawn from 31 datasets, including 28 existing datasets and three newly constructed collections: IQTest, FunctionQA, and PaperQA. Its visual contexts include geometry diagrams, function plots, scientific figures, and visual puzzles. The benchmark covers textbook question answering, visual question answering, geometry problem solving, math word problems, and figure question answering, providing a diverse assessment of mathematical reasoning grounded in visual inputs.
- **We-Math** (Qiao et al., 2025) contains approximately 6,500 visual mathematics problems organized around 67 knowledge concepts across five levels of granularity. It decomposes composite problems into subproblems associated with the required concepts, allowing the benchmark to distinguish knowledge deficiencies from failures to combine or generalize that knowledge. Its diagnostic framework examines insufficient knowledge, inadequate generalization, complete mastery, and rote memorization beyond final-answer correctness.
- **HallusionBench** (Guan et al., 2024) evaluates image-context reasoning with an emphasis on language hallucination and visual illusion. It comprises 346 images and 1,129 questions constructed by human experts. Related questions and visual control groups probe whether responses remain consistent with the actual image content, allowing analysis of failures caused by misleading visual cues or language-based expectations.

[1] https://huggingface.co/datasets/hiyouga/geometry3k

- **ChartQA** (Masry et al., 2022) evaluates question answering that combines chart interpretation with numerical and logical reasoning. The dataset includes approximately 9,600 human-written questions and 23,100 questions generated from human-written chart summaries. Answering these questions requires identifying relevant chart elements and relating their values, often through arithmetic or comparison. We use the official evaluation data for the broader comparison.
- **MMMU** (Yue et al., 2024) evaluates multimodal understanding and reasoning that require college-level subject knowledge. It covers 30 subjects and 183 subfields, with heterogeneous visual inputs such as charts, diagrams, maps, tables, musical notation, and chemical structures. The questions test the integration of specialized knowledge with visual evidence across disciplines. We report results on the validation set because the test-set ground-truth labels are not publicly available.

System Prompt: Universal Multimodal Reasoning

You are a helpful assistant solving multimodal reasoning problems. Use the image, question, and options when provided to identify the relevant visual evidence and derive the answer through any necessary recognition, comparison, calculation, or reasoning. Use domain knowledge only when necessary, and do not assume visual facts unsupported by the image or question. Keep the reasoning concise and directly relevant to the answer.
Output exactly two XML elements and nothing else: first a `<think>` element closed with `</think>`, then an `<answer>` element closed with `</answer>`. The `<answer>` element must contain only the final answer in the format required by the question. Place no text before `<think>` and no text after `</answer>`.

Training Details. Table 5 summarizes the complete training and skill-library configurations. For the controlled self-improvement comparison, we follow the VISTA protocol (Zhong et al., 2026) and use Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct as the base models. Specifically, we adopt the same rollout budget, sampling configuration, number of self-improvement iterations, and SFT schedule as VISTA (Zhong et al., 2026). Following its Qwen2.5-VL setting, only the LLM backbone is updated during SFT, while the vision encoder and multimodal projector remain frozen. Training is implemented with LLaMA-Factory[2]. This controlled setting is used for the comparison in Table 1, where Skill-π-SFT excludes subsequent RL optimization.

For the full Skill-π, we initialize the policy on Vision-SR1-Cold-9K and perform subsequent RL post-training on Vision-SR1-47K (Li et al., 2026b). The cold-start stage uses full-parameter fine-tuning, whereas RL is implemented with VERL[3] and vLLM for on-policy rollout generation, with LoRA applied to the attention projection layers. The shared policy is optimized using the stage-masked GRPO objective described in Sec. 3.3. For skill-guided interaction, semantic retrieval first returns a candidate skill set after visual-state filtering, from which the policy selects two distinct skills to form two branches. Each branch selects one skill and performs multiple utilization rollouts under the fixed skill-conditioned context. The resulting branch structure provides the comparison groups used for stage-specific selection and utilization credit. The same cross-sample grouping criterion is used for initial skill construction and subsequent library consolidation. Skill utility is initialized and updated online according to the empirical reuse statistics described in Sec. C.4. All numerical configurations are reported in Table 5.

Evaluation Details. We use MLLMEvalKit[4] for model evaluation and follow the official evaluation protocol of each benchmark. For the controlled comparison on SLAKE, VQA-RAD, and Geometry3K, we follow VISTA and evaluate all models in a zero-shot manner on the corresponding test sets using greedy decoding. For the broader benchmark evaluation, we use the official evaluation split of each dataset, except for MMMU where the validation split is used. Benchmark-specific answer extraction, normalization, and scoring are handled by MLLMEvalKit and the corresponding official evaluators.

[2] https://github.com/hiyouga/LlamaFactory
[3] https://github.com/volcengine/verl
[4] https://github.com/open-compass/MLLMEvalKit

Table 5: Training and skill-library hyperparameters.

Stage	Hyperparameter / setting	Value
SFT-only	Base models	Qwen2.5-VL-3B/7B-Instruct
	Self-improvement iterations	3
	Training epochs per iteration	3
	Reasoning trajectories per query	10
	Sampling temperature	1.0
	Maximum response length	2,048 tokens
	Global batch size	32
	Peak learning rate	1×10^{-5}
	Maximum image resolution	512×512
	Parameter update	LLM backbone only
Cold-start	Training data	Vision-SR1-Cold-9K
	Parameter update	Full fine-tuning
	Optimizer	AdamW
	Training epochs	3
	Peak learning rate	1×10^{-5}
	Weight decay	0.01
	Warmup ratio	0.05
	Per-device batch size	1
	Gradient accumulation steps	16
	Maximum response length	2,048 tokens
RL	Training data	Vision-SR1-47K
	Parameter update	LoRA
	LoRA rank	64
	LoRA scaling factor	128
	LoRA dropout	0.05
	LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`
	Learning rate	1×10^{-6}
	Maximum prompt length	2,048 tokens
	Maximum response length	8,192 tokens
	GRPO clipping ratio	0.20
	KL coefficient	$\beta = 0.001$
	Selection objective weight	$\lambda_{\text{sel}} = 0.3$
	Evolution objective weight	$\lambda_{\text{evo}} = 0.5$
Skills	Question-grouping similarity threshold	$\tau_{\text{grp}} = 0.80$
	Skill-selection branches per query	$K = 2$
	Utilization rollouts per branch	$G = 8$
	Minimum unique samples per construction group	$m = 5$
	Minimum unique samples per evolution group	5
	Refinement similarity threshold	0.80
	Initial empirical utility	$u_0 = 0.50$
	Utility EMA coefficient	$\alpha = 0.10$
	Retrieved candidate skills per query	5
	Total utilization rollouts per query	$KG = 16$
Hardware	Numerical precision	bfloat16
	GPU configuration	8 NVIDIA A800 GPUs; 80 GB each

The grouping threshold and minimum support apply to both initial skill construction and later consolidation. Each query selects two distinct skills and uses eight utilization rollouts per branch, for 16 rollouts in total.

B Additional Analysis

Metric Definitions. The average score in Tables 6 and 7 evaluates the final model on the same seven benchmarks as Table 2:

$$\text{Avg.Acc.} = \frac{1}{D}\sum_{d=1}^{D}\text{Acc}_d, \qquad D = 7. \tag{8}$$

The benchmarks are MathVerse, MATH-Vision, MathVista, We-Math, HallusionBench, ChartQA, and MMMU. The remaining metrics assess skill-guided interaction on an analysis set of N_{ana} queries, rather than the final-model benchmark average. Rates below are multiplied by 100 when reported as percentages.

Guided Success is query-level success under the full $KG = 16$ rollout budget, with $K = 2$ selected skills and $G = 8$ trajectories per branch:

$$S_i^{\text{guided}} = \mathbb{I}\left[\max_{k,g} r_{i,k,g} = 1\right], \qquad \text{GuidedSuccess} = \frac{1}{N_{\text{ana}}}\sum_{i=1}^{N_{\text{ana}}} S_i^{\text{guided}}. \tag{9}$$

Its denominator includes the entire analysis set, unlike the recovery metric in Figure 4, which uses an initial $0/5$ unresolved subset and five guided rollouts.

Selection Quality measures whether the two selected skills include a downstream-optimal candidate from the retrieved top five. Evaluating every candidate with $G_{\text{eval}} = G = 8$ rollouts defines

$$\begin{aligned} q_{i,j} &= \frac{1}{G_{\text{eval}}}\sum_{g} r_{i,j,g}^{\text{eval}}, \quad \mathcal{B}_i^\star = \arg\max_{j\in\mathcal{B}_i} q_{i,j}, \\ \text{SelectionQuality} &= \frac{1}{N_{\text{ana}}}\sum_{i} \mathbb{I}[\mathcal{K}_i^{\text{selected}} \cap \mathcal{B}_i^\star \neq \varnothing]. \end{aligned} \tag{10}$$

Here $\mathcal{K}_i^{\text{selected}}$ contains the selected candidate IDs; all tied maxima count as optimal. This top-two hit rate requires evaluation of all five candidates, beyond the two selected branches.

Harm Rate is the fraction of unguided-solvable queries for which all guided rollouts fail. With S_i^{base} denoting unguided success on the same query and policy checkpoint,

$$\text{HarmRate} = \frac{\sum_i \mathbb{I}[S_i^{\text{base}} = 1 \wedge S_i^{\text{guided}} = 0]}{\sum_i \mathbb{I}[S_i^{\text{base}} = 1]}. \tag{11}$$

The unguided condition uses the same total budget of 16 rollouts, with success defined by at least one correct response. If no unguided queries succeed, this conditional rate is undefined.

B.1 Analysis of Credit Assignment

We further examine whether the stage-specific credit assignment in Skill-π is necessary beyond applying a standard trajectory-level GRPO objective. Table 6 compares our design with three alternatives. Flat Trajectory-Level GRPO places all skill-conditioned trajectories of the same task into a single comparison group, without distinguishing skill selection from skill utilization. Shared Task-Level Advantage uses the same task-level learning signal for both stages, while Utilization-Only GRPO optimizes skill-conditioned reasoning without assigning an explicit selection credit.

Stage-Specific Credit achieves the highest average accuracy of 65.5%, compared with 63.6%, 64.2%, and 63.9% for the three alternatives, respectively. It also yields the highest guided success rate and selection quality. These results indicate that directly comparing trajectories generated under different skill decisions introduces a less precise learning signal. Separating branch-level skill selection from within-branch skill utilization provides more informative credit for the corresponding decisions.

B.2 Analysis of Skill Retrieval and Selection

We next evaluate how the retrieved skill candidates should be selected before skill-conditioned reasoning. Table 7 compares the learned policy selection in Skill-π with random selection, direct Top-1

Table 6: Comparison of different credit assignment strategies. Stage-specific credit separates the learning signals for skill selection and skill utilization according to their corresponding downstream outcomes.

Credit Assignment	Avg. Acc.	Guided Success	Selection Quality
Flat Trajectory-Level GRPO	63.6	58.9	61.7
Shared Task-Level Advantage	64.2	60.1	63.4
Utilization-Only GRPO	63.9	60.5	60.8
Stage-Specific (Ours)	**65.5**	**62.8**	**66.1**

Table 7: Effect of skill retrieval and selection strategies. Learned skill selection improves downstream performance while reducing harmful use of retrieved skills.

Selection Strategy	Avg. Acc.	Guided Success	Harm Rate
Random Skill	61.8	54.6	8.7
Top-1 Similarity	63.4	58.2	6.3
w/o Visual-State Filter	64.1	59.7	5.5
Policy Selection (Ours)	**65.5**	**62.8**	**3.2**

similarity retrieval, and a variant without visual-state filtering. Random Skill samples one skill from the retrieved candidates, while Top-1 Similarity directly uses the candidate with the highest semantic similarity without policy-based applicability judgment. All strategies use 16 utilization rollouts per query; one-skill baselines allocate the full budget to that skill. The variant without visual-state filtering performs semantic retrieval and policy selection over the full skill library.

Direct similarity retrieval improves over random skill selection, increasing the average accuracy from 61.8% to 63.4%, but remains below the 65.5% achieved by Policy Selection. Removing visual-state filtering reaches 64.1%, showing that learned selection can reject some semantically similar but structurally less compatible skills, while the explicit visual-state constraint further improves candidate quality. Policy Selection also reduces the harm rate from 8.7% under random selection and 6.3% under Top-1 similarity to 3.2%. This result suggests that semantic similarity alone is insufficient to determine skill applicability, and that combining structural compatibility with learned selection leads to more reliable skill reuse.

B.3 Analysis of Capability Expansion and Skill Utility

Figure 4 examines whether the retrieved skills function as reusable cross-instance experience rather than only as descriptions of their source trajectories. On the subset for which all five initial unguided rollouts failed, skill-guided sampling recovered 29.53%, 25.39%, and 22.46% of queries on SLAKE, VQA-RAD, and Geometry3K, respectively, corresponding to an unweighted average of 25.79% (Figure 4a). Because the unguided value is fixed at zero by the subset definition, this result should be interpreted as conditional recovery under the same five-rollout budget, not as an independent unguided reevaluation or an estimate of the overall accuracy gain.

The utility analysis provides a complementary view of reuse reliability. Using the utility value available before each reuse avoids incorporating the outcome being evaluated into its predictor. The subsequent success rate increases monotonically from 38.4% in the lowest-utility bucket to 76.3% in the highest-utility bucket (Figure 4b). This trend is consistent with interpreting u_j as an empirical transferability signal and supports retaining reuse outcomes in skill evolution and library consolidation. Together with the selection results in Table 7, these findings indicate that reliable reuse depends both on selecting a structurally compatible candidate and on preserving strategies with favorable downstream histories. Since only aggregate bucket rates are available, however, the utility trend establishes an association and does not by itself demonstrate causality or statistical significance.

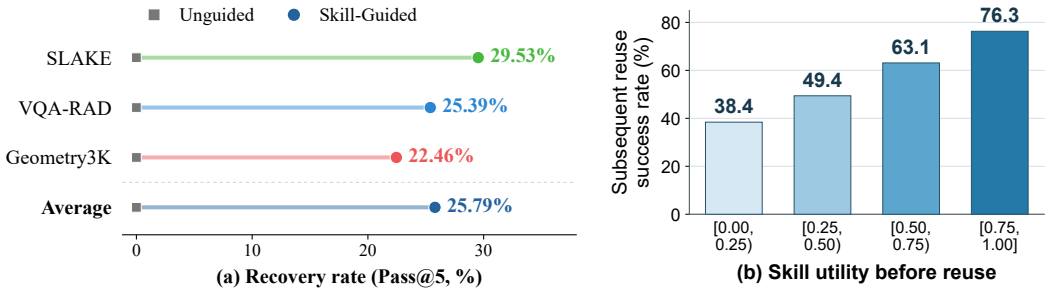


Figure 4: Complementary evidence for cross-instance skill reuse. **(a)** Recovery of previously unresolved queries. For each dataset, we select queries for which all five initial unguided rollouts are incorrect and report the fraction with at least one correct response in five skill-guided rollouts. Average denotes the unweighted mean across datasets; the unguided rate is zero by subset construction. **(b)** Relationship between skill utility and subsequent reuse success. Skills are grouped by the utility recorded before reuse, and the success rate is measured on the subsequent interactions.

Table 8: Computation resources across self-improvement iterations. Total GPU-h sums rollout, GRPO, and skill operations.

Round	Avg. rollout length	Rollout GPU-h	GRPO GPU-h	Skill operations GPU-h	Total GPU-h
Iter. 1	735 tokens	37.6	19.2	6.4	**63.2**
Iter. 2	692 tokens	35.2	18.6	6.8	**60.6**
Iter. 3	648 tokens	33.4	18.0	7.2	**58.6**

B.4 Computation Resource Analysis

Table 8 summarizes the computational cost of the three self-improvement iterations. From the first to the third iteration, average rollout length decreases by 11.8%, accompanied by reductions of 11.2% in rollout GPU-hours and 6.3% in GRPO GPU-hours. Although skill operations become 12.5% more expensive, they account for only 10.1%–12.3% of each iteration's total GPU-hours. The reductions in rollout and optimization costs therefore outweigh the additional skill-processing cost, lowering total GPU-hours by 7.3%. These measurements show that maintaining and evolving the skill library does not lead to increasing total cost across the reported iterations; however, they do not isolate whether skill reuse itself causes the shorter rollouts.

B.5 Hyperparameter Sensitivity

Figure 5 examines four hyperparameters governing skill construction and reinforcement learning. For question grouping, $\tau_{\text{grp}} = 0.80$ improves the average score by 1.2% and 1.1% over the lowest and highest tested thresholds, respectively. Similarly, $m = 5$ improves over singleton groups by 1.7%, while increasing m to ten reduces the score by 1.0%. These patterns are consistent with a trade-off between grouping compatible experiences and retaining enough examples to extract reusable strategies. For reward weighting, $\lambda_{\text{sel}} = 0.3$ and $\lambda_{\text{evo}} = 0.5$ yield the highest scores in their respective sweeps, improving over the lowest tested weights by 0.9% and 1.4%. Further increasing either weight reduces performance, suggesting that stronger auxiliary incentives do not necessarily improve task learning. Across all four sweeps, the nearest tested settings on either side of the peak remain within 0.7% of the best score, indicating moderate local sensitivity. All differences here denote absolute changes on the percentage scale.

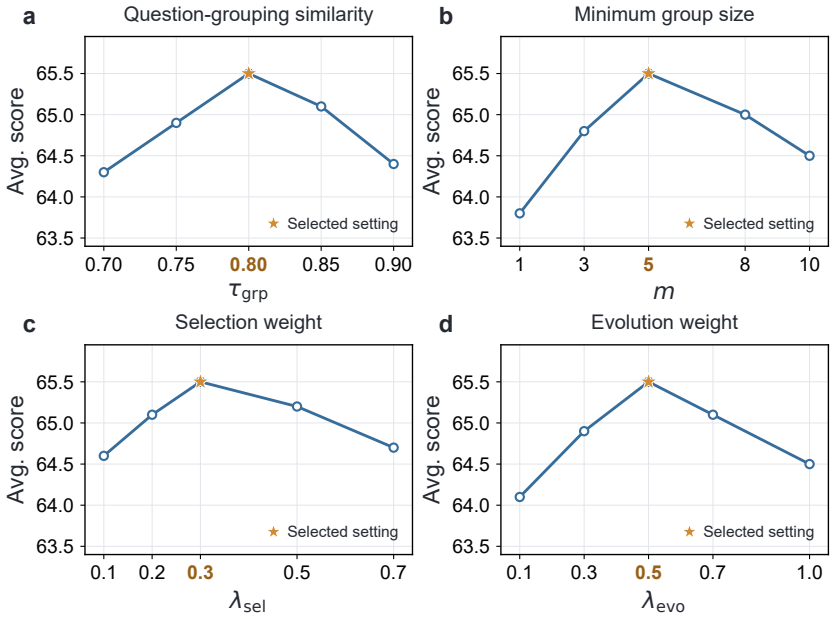


Figure 5: Hyperparameter sensitivity of Skill-π. Average score (%) as a function of **(a)** question-grouping similarity, **(b)** minimum group size, **(c)** selection weight, and **(d)** evolution weight. Stars mark the selected settings; all panels share the same vertical scale.

C Skill Construction Details

C.1 Visual State.

Skill-π associates each sample with a discrete visual state $v_i = (\text{domain}_i, \text{pattern}_i, \text{operation}_i)$ to provide a coarse characterization of the conditions under which a reasoning skill may be applicable. The visual state is inferred from the image–question pair by the shared MLLM using a fixed taxonomy. It consists of three complementary attributes:

- **Visual Domain.** The visual content is categorized as *mathematical*, *chart/document*, *medical*, or *general*. The first three correspond to visual mathematics and diagrams, charts or document-style visual content, and medical or biomedical images, respectively, while *general* covers the remaining visual content.
- **Task Pattern.** The question is categorized as *identify*, *classify*, *compare*, *count*, *measure*, or *solve*. These labels respectively describe tasks that ask what is present, assign a category or property, compare visible entities or quantities, count visible instances, directly read or estimate a numerical value, or derive an answer through calculation, rules, theorems, or multistep inference.
- **Operation Type.** The required operation is categorized as either *perception* or *reasoning*. *Perception* denotes tasks that can be resolved mainly by directly reading, detecting, recognizing, or extracting visual information, whereas *reasoning* requires further transformation of the visual evidence through comparison, calculation, logical relations, rules, or multistep inference.

When multiple task patterns appear plausible, we apply a fixed precedence rule to obtain a unique canonical label. *Solve* is used when explicit calculation, a theorem, a rule, or multistep inference is required; *compare* is used when comparison itself resolves the task; *measure* is reserved for values that can be directly read or visually estimated, whereas calculated values are categorized as *solve*; *count* is used when counting visible instances directly answers the question; *classify* is used for category or property assignment; and *identify* covers tasks asking what is present, where it is, or whether it exists. The operation type is determined independently of the task pattern. For example, a comparison can be categorized as *perception* when the relevant values are directly visible, or as *reasoning* when additional inference is required.

System Prompt: Visual State Assignment

Analyze the image and question and assign exactly one canonical label for each visual task attribute. Choose labels only from the allowed sets below. Do not create new labels or variants.
visual_domain:
mathematical: visual mathematics, geometry, equations, diagrams, or quantitative mathematical problems
chart_document: charts, plots, tables, diagrams, forms, or document-style visual content
medical: medical images, clinical figures, pathology, radiology, or biomedical visual content
general: all other visual content
task_pattern:
identify: determine what visible entity, object, text, relation, or finding is present, or whether it exists
classify: assign a visible entity or scene to a category, property, condition, shape, color, or class
compare: compare two or more visible entities, quantities, or properties
count: determine the number of visible items
measure: directly read or estimate a numeric value from the visual content
solve: derive the answer through a theorem, formula, calculation, symbolic manipulation, or multistep inference
operation_type:
perception: the answer can be obtained mainly by directly reading, detecting, recognizing, or extracting information from the image
reasoning: the visual evidence must be transformed through comparison, calculation, a rule, theorem, logical relation, or multistep inference
Return exactly three XML elements: `<visual_domain> </visual_domain>`, `<task_pattern> </task_pattern>`, and `<operation_type> </operation_type>`.

The visual state is used as a structural compatibility constraint during skill construction and retrieval. Successful samples are first separated according to their visual states before semantic grouping, and skill retrieval is restricted to structurally compatible skills before question-semantic matching.

C.2 Group-Level Skill Construction.

Given the verified successful trajectory pool $\mathcal{T}^{(0)}$, we construct skills through cross-sample abstraction rather than independently distilling one skill from each successful trajectory. For each successful sample x_i, we first obtain its question embedding $e_i = E_q(q_i)$ using the frozen text encoder E_q. Samples are grouped only when they share the same visual state and are sufficiently similar in question semantics:

$$v_i = v_{i'}, \qquad \cos(e_i, e_{i'}) \geq \tau_{\text{grp}}, \tag{12}$$

This first-stage constraint ensures that skill induction is performed over problems with compatible visual structures and related reasoning requirements. For each verified successful trajectory $\tau_{i,k}$ in a retained group, the shared MLLM first extracts a transient local observation

$$\ell_{i,k} = F_{\text{local}}\left(x_i,\, v_i,\, \tau_{i,k}\right) = \left(b_{i,k},\, p_{i,k}\right), \tag{13}$$

where $b_{i,k}$ denotes the reasoning behavior associated with the successful trajectory and $p_{i,k}$ denotes its sample-independent transferable principle. During this extraction, sample-specific entities, values, labels, answer choices, and final answers are explicitly removed. The resulting $\ell_{i,k}$ serves only as intermediate evidence and is not stored as a persistent skill. For each group $\mathcal{G}_j$, local evidence from different samples is then jointly abstracted by the shared MLLM:

$$(\delta_j, \rho_j) = F_{\text{group}}\left(v_j,\, \{\ell_{i,k}\}_{i \in \mathcal{G}_j}\right), \qquad |\mathcal{U}(\mathcal{G}_j)| \geq m, \tag{14}$$

where $\mathcal{U}(\mathcal{G}_j)$ denotes the set of unique samples represented in the group, δ_j describes the transferable capability and its applicability, and ρ_j specifies the corresponding reasoning strategy. If no sufficiently consistent reasoning behavior is supported across the group, the induction process returns `<NO_SKILL/>` and no skill is added to the library. This group-level construction explicitly separates sample-level evidence extraction from persistent skill induction. Consequently, a successful trajectory alone cannot directly produce a skill; a reasoning behavior must recur across multiple related samples before being retained as reusable experience.

System Prompt: Local Success Evidence

You analyze one verified successful trajectory for a single visual reasoning problem.
Your goal is NOT to solve the problem and NOT to create a Skill. Identify the reusable reasoning behavior that made the trajectory successful. Remove sample-specific entities, values, labels, answer choices, and final answers. Do not introduce facts, rules, or domain knowledge unsupported by the trajectory.
Return exactly:
<successful_behavior>
The key visual reasoning behavior that contributes to the successful trajectory.
</successful_behavior>
<transferable_principle>
Express the useful behavior without sample-specific entities, values, answer choices, or final answers.
</transferable_principle>
Output no additional text.

System Prompt: Group-Level Skill Induction

You are given reasoning evidence from several DIFFERENT visual reasoning problems belonging to a similar problem family.
The problems may involve mathematical reasoning, charts and documents, general visual reasoning, or medical visual reasoning.
Each evidence item summarizes:
- the behavior associated with one verified successful trajectory;
- a sample-level transferable principle.

Your task is to induce a SINGLE transferable Skill that captures a reasoning behavior shared across the different samples.
The Skill must satisfy all of the following:
1. be supported by successful evidence from different samples;
2. remain useful when sample-specific objects, values, labels, answer choices, and final answers change;
3. describe a reusable visual reasoning capability, not an individual solution;
4. be specific enough to alter reasoning behavior;
5. avoid generic advice such as "analyze carefully", "inspect the image", or "reason step by step";
6. introduce no facts, rules, or domain knowledge unsupported by the evidence.

If no sufficiently consistent transferable behavior exists across the group, return exactly:
<NO_SKILL/>
Otherwise return exactly:
<description>
Describe what capability the Skill provides and under what observable conditions it is applicable.
</description>
<strategy>
Give a concise ordered procedure for applying the Skill.
</strategy>
Output no additional text.

C.3 Skill Selection and Utilization Details

Skill Retrieval. Given a training sample $x_i = (I_i, q_i)$ with visual state v_i, we first restrict the current skill library $\mathcal{S}^{(t)}$ to structurally compatible skills:

$$\mathcal{C}_i^{(t)} = \left\{ s_j \in \mathcal{S}^{(t)} \mid v_j = v_i \right\}. \tag{15}$$

The shared MLLM then generates a concise retrieval query z_i describing the visual reasoning capability required by the current problem without solving it. Using a frozen text encoder $E(\cdot)$, we retrieve the candidate set

$$\mathcal{B}_i^{(t)} = \mathrm{Top}_{L, s_j \in \mathcal{C}_i^{(t)}} \cos\left(E(z_i), E(\delta_j)\right), \tag{16}$$

where δ_j is the description of skill s_j. We retain the top-L candidates with $L = 5$ in all experiments. The visual-state constraint is applied before semantic retrieval so that question-level similarity alone cannot introduce structurally incompatible skills.

System Prompt: Retrieval Query Generation

Summarize the visual reasoning capability required by the image and question for retrieval. Describe what needs to be recognized or reasoned about and the operation required, without solving the problem or including the answer. Return only one concise retrieval query.

Skill Selection. Rather than directly using the highest-similarity retrieved skill, Skill-π lets the shared policy judge its actual applicability to the current image–question pair. At decision k, let $\mathcal{B}_{i,k}^{(t)}$ be the retrieved set with previously selected skill IDs removed. The policy samples K decisions:

$$d_{i,k} \sim \pi_\theta^{(t)}\left(\cdot \mid x_i, v_i, \mathcal{B}_{i,k}^{(t)}\right), \qquad k = 1, \ldots, K, \tag{17}$$

where each decision $d_{i,k}$ contains a ranking over the remaining candidates and either selects the highest-ranked applicable skill $s_{i,k}^*$ or rejects all candidates. Previously selected skill IDs are excluded from subsequent selections for the same query. A rejected selection is represented as $s_{i,k}^* = \varnothing$ and defines an unguided reasoning branch.

This policy-based selection provides a second applicability check beyond embedding similarity. In particular, the policy considers whether the candidate skill matches the current visual configuration, required givens, and reasoning operation, rather than treating semantic similarity as sufficient evidence of applicability.

System Prompt: Skill Selection

Rank the candidate Skill cards for the current image and question.
Treat every Skill as optional advice, not ground truth. Semantic similarity alone is not enough: a Skill's visual configuration, required givens, and reasoning operation must match the current problem. Use the image, question, and initial visual state when comparing Skills. Do not solve the question.
Return exactly one JSON object with this schema:
`{"ranking": ["skill_id", "..."], "selected_skill_id": "skill_id or null"}`
Rules:

- `ranking` must contain every provided `skill_id` exactly once, ordered from most to least useful;
- `selected_skill_id` must equal the first ranked ID only when it is genuinely applicable;
- if none is genuinely applicable, `selected_skill_id` must be `null`;
- output no other keys or text.

Skill Utilization. Each selection decision defines an independent reasoning branch. For a selected skill $s_{i,k}^*$, we provide only its description $\delta_{i,k}^*$ and strategy $\rho_{i,k}^*$ as optional guidance, and sample G reasoning trajectories:

$$\tau_{i,k,g} \sim \pi_\theta^{(t)}\left(\cdot \mid x_i, v_i, s_{i,k}^*\right), \qquad g = 1, \ldots, G. \tag{18}$$

If $s_{i,k}^* = \varnothing$, the same policy instead solves the problem directly without external skill guidance. Each trajectory is evaluated using the task verifier,

$$r_{i,k,g} = \mathcal{V}\left(\mathrm{Ans}(\tau_{i,k,g}), a_i\right), \tag{19}$$

where $r_{i,k,g} \in \{0, 1\}$ denotes answer correctness.

System Prompt: Skill Utilization

When a selected Skill is provided, treat it as optional advice rather than ground truth. Check it against the image and question, and ignore it when it conflicts with visible evidence. Do not mention or discuss the Skill in the response.
When the selected Skill is null, solve directly from the image, question, and options when provided.

Importantly, a selected skill is treated only as optional reasoning guidance. The policy is explicitly instructed to check the skill against the visual evidence and ignore it when the suggested strategy conflicts with the current problem. This design reduces negative transfer from imperfect or only partially applicable skills.

Branch Structure. We distinguish three quantities in skill-guided interaction. The retrieval stage retains five candidate skills in $\mathcal{B}_i^{(t)}$; the policy then samples K skill-selection decisions conditioned on this same candidate set; and each resulting selection branch generates G utilization trajectories. Therefore, the candidate-set size controls retrieval coverage, K controls exploration over alternative skill choices, and G controls stochastic exploration under a fixed skill-conditioned branch.

C.4 Skill Evolution Details

Experience Reflection. Skill evolution starts from the verified interaction experience generated during skill-guided exploration. For each task x_i, we collect the outcomes of all selection branches and their corresponding utilization rollouts as

$$\mathcal{O}_i = \left\{ \left(s_{i,k}^*, \tau_{i,k,g}, r_{i,k,g} \right) \right\}_{k=1,g=1}^{K,G}. \tag{20}$$

Successful trajectories are first summarized into local reusable evidence using the same abstraction principle adopted during initial skill construction. These trajectory-level observations capture the successful reasoning behavior without directly modifying the persistent skill library.

The shared policy then reflects over the complete set $\mathcal{O}_i$ to compare alternative skill selections, successful reasoning patterns, remaining failure modes, and uncovered skill gaps. We denote this task-level evolution proposal as

$$\tilde{s}_i = F_{\text{evo}}\left(x_i,\, v_i,\, \mathcal{O}_i\right), \tag{21}$$

where $\tilde{s}_i$ summarizes whether the observed experience supports reinforcing an existing reasoning behavior, revising an incomplete strategy, or expanding the library with a previously uncovered capability. The proposal itself is transient and does not immediately modify the skill library. Failed trajectories are used only to identify limitations and skill gaps and are never treated as successful evidence.

System Prompt: Skill Evolution Reflection

Reflect on the complete group of verified attempts for one task. Compare skill selections, successful reasoning patterns, remaining failure modes and skill gaps. Decide whether to reinforce, revise or expand a reusable skill. Do not treat failed behavior as a success.
Return exactly `<successful_behavior>...</successful_behavior>` followed by `<transferable_principle>...</transferable_principle>`.
Use the second field for the transferable principle, limitations, skill gap and proposed revision. If none succeeded, state that explicitly; do not invent successful evidence. This is a proposal only.

Utility Update and Delayed Credit. One utility update corresponds to one subsequent query on which a constructed skill is used. Its $G = 8$ rollout outcomes are averaged before updating the EMA:

$$\bar{r}_j^{(\ell)} = \frac{1}{G}\sum_{g=1}^{G} r_{j,\ell,g}, \qquad u_j^{(\ell)} = (1-\alpha)u_j^{(\ell-1)} + \alpha \bar{r}_j^{(\ell)}, \quad u_j^{(0)} = u_0. \tag{22}$$

Here ℓ counts skill–query reuse interactions, not individual trajectories; `num_uses` increases once per such interaction. We use $u_0 = 0.5$ and $\alpha = 0.10$. Source-task rewards determine successful construction evidence, whereas subsequent-query branch means determine utility.

Let $j(i)$ map a stored task-level proposal to the skill constructed from its group. The target $\hat{u}_i = u_{j(i)}$ is shared by all proposals contributing to that skill. At an update, $\mathcal{H}$ contains proposals linked to constructed skills with reuse feedback collected after their construction or integration, drawn across different construction groups. Proposal indices identify stored task–round records. We normalize their targets as

$$A_i^{\text{evo}} = \frac{\hat{u}_i - \mu_{\mathcal{H}}^{\text{evo}}}{\sigma_{\mathcal{H}}^{\text{evo}} + \eta}, \quad \mu_{\mathcal{H}}^{\text{evo}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \hat{u}_i, \quad (\sigma_{\mathcal{H}}^{\text{evo}})^2 = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} (\hat{u}_i - \mu_{\mathcal{H}}^{\text{evo}})^2. \tag{23}$$

This compares task-group proposals across construction groups, rather than comparing trajectories within one task. Proposals awaiting sufficient support or subsequent feedback are retained without an evolution update. Equal target utilities yield zero relative credit.

Skill Library Consolidation. The persistent proposal buffer accumulates task-level records with at least one successful source rollout:

$$\mathcal{E}^{(t)} = \{(x_i, \mathcal{O}_i, \tilde{s}_i) \mid \text{Success}(\mathcal{O}_i) = 1\}, \quad \text{Success}(\mathcal{O}_i) = \mathbb{I}[\max_{k,g} r_{i,k,g} = 1]. \tag{24}$$

Records are grouped by visual state and question similarity. A group is processed once it contains at least m unique samples: the same group-level skill-construction procedure extracts recurring successful evidence and either returns a reusable skill or <NO_SKILL/>. A constructed skill is added to the library or integrated with a compatible entry; its source proposal links are retained through refinement or merging. Insufficient-support groups remain buffered. Thus, no prior reuse utility is required to construct a new skill. The resulting library update is

$$\mathcal{S}^{(t+1)} = \text{Consolidate}(\mathcal{S}^{(t)}, \mathcal{E}^{(t)}). \tag{25}$$

Once the skill is reused on subsequent queries, its utility supplies the target for all associated proposal-generation records. We retain each record's generated tokens and behavior-policy log probabilities for its later stage-masked update.

Evolution Configuration. The grouping, minimum-support, refinement, and utility-update settings are summarized in Table 5. Skill evolution reuses the grouping and minimum-support constraints of initial construction, while the refinement similarity threshold governs updates to compatible existing skills.

D Algorithm Details

Overall Algorithm. Algorithm 1 summarizes the T evolution rounds of Skill-π. Starting from the current policy and persistent skill library, the policy first performs skill-guided interaction and one reflection per task group. Buffered proposals undergo cross-sample skill construction; subsequent skill reuse returns credit to their source reflections for stage-specific policy optimization.

D.1 From GRPO to Stage-Specific Credit Assignment

Standard group-relative credit. Outcome-supervised GRPO (Shao et al., 2024c) compares responses to the same query and broadcasts each response's normalized reward to its generated tokens. For the KG skill-conditioned trajectories of sample i, a flat version uses

$$\mu_i = \frac{1}{KG}\sum_{k,g} r_{i,k,g}, \qquad (\sigma_i^{\text{flat}})^2 = \frac{1}{KG}\sum_{k,g}(r_{i,k,g} - \mu_i)^2, \qquad A_{i,k,g}^{\text{flat}} = \frac{r_{i,k,g} - \mu_i}{\sigma_i^{\text{flat}} + \eta}. \tag{26}$$

Here $\eta > 0$ is the numerical stabilizer and ϵ is the clipping radius, following the main-text notation. A flat group mixes variation in the selected skill with variation in its execution. We retain GRPO's clipped surrogate but change the comparison groups and the tokens receiving each credit signal.

Separating selection from utilization. Fix a query and candidate library, and suppress the sample index temporarily. Let d be the generated selection decision, $s(d)$ its selected skill, and τ a utilization trajectory. The shared policy factorizes as $p_\theta(d, \tau) = \pi_\theta(d \mid x)\pi_\theta(\tau \mid x, s(d))$, where fixed stage prompts and retrieved candidates are implicit. Define $Q_\theta(d) = \mathbb{E}[r \mid d]$ and $V_\theta = \mathbb{E}_d[Q_\theta(d)]$. The score-function identity gives

$$\begin{aligned}\nabla_\theta \mathbb{E}[r] &= \mathbb{E}[r\nabla_\theta \log \pi_\theta(d \mid x) + r\nabla_\theta \log \pi_\theta(\tau \mid x, s(d))] \\ &= \mathbb{E}[(Q_\theta(d) - V_\theta)\nabla_\theta \log \pi_\theta(d \mid x) + (r - Q_\theta(d))\nabla_\theta \log \pi_\theta(\tau \mid x, s(d))]\,.\end{aligned} \tag{27}$$

The second equality follows from conditional expectation and the zero mean of each conditional policy score. Thus selection receives the expected value of its branch, whereas utilization receives the residual relative to that branch. This identity holds for the population, unnormalized task-reward gradient; sample-based normalization and clipping below define a surrogate update.

The quantities μ_i and $\bar{r}_{i,k}$ below equal the main-text μ_i^{sel} and $\mu_{i,k}$, respectively. Under the rollout policy, estimate $Q(d_k)$ by $\bar{r}_{i,k} = G^{-1}\sum_g r_{i,k,g}$ and V by $\mu_i = K^{-1}\sum_k \bar{r}_{i,k}$. Every sampled centered reward then decomposes *exactly* into two terms:

$$r_{i,k,g} - \mu_i = \underbrace{\bar{r}_{i,k} - \mu_i}_{\text{between-branch selection credit}} + \underbrace{r_{i,k,g} - \bar{r}_{i,k}}_{\text{within-branch utilization credit}}. \tag{28}$$

Algorithm 1 Skill–Policy Co-Evolution in Skill-π

Require: Training data $\mathcal{D}$, policy $\pi_\theta^{(0)}$, library $\mathcal{S}^{(0)}$, rounds T
Ensure: Policy $\pi_\theta^{(T)}$ and library $\mathcal{S}^{(T)}$

Initialize persistent proposal buffer and proposal-to-skill links
for $t = 0, \ldots, T-1$ **do**
  # Group-Level Interaction and Reflection
  for each $(x_i, a_i) \in \mathcal{D}$, where $x_i = (I_i, q_i)$ **do**
    Infer v_i and retrieve $L = 5$ candidate skills
    Select $K = 2$ distinct skills to form utilization branches
    for $k = 1, \ldots, K$ **do**
      Sample $G = 8$ trajectories $\{\tau_{i,k,g}\}_g$ and verify $\{r_{i,k,g}\}_g$
      Update the reused skill's utility once using $\bar{r}_{i,k} = G^{-1} \sum_g r_{i,k,g}$
    end for
    Collect all KG outcomes as $\mathcal{O}_i$; generate one proposal $\tilde{s}_i$
    Store proposal tokens and their behavior-policy log probabilities
    Add successfully supported $(x_i, \mathcal{O}_i, \tilde{s}_i)$ to the proposal buffer
  end for
  # Skill Construction and Delayed Feedback
  Group buffered proposals by visual state and question similarity
  for each group with at least m unique samples **do**
    Apply group-level skill construction; integrate a valid skill into the library
    Link all contributing proposals to the constructed or integrated skill
  end for
  Retain insufficient-support groups; obtain $\mathcal{S}^{(t+1)}$
  Form $\mathcal{H}$ from linked proposals with subsequent reuse feedback
  # Stage-Specific Policy Optimization
  Compute $A^{\text{util}}_{i,k,g}$ within branches and $A^{\text{sel}}_{i,k}$ across branches
  If $\mathcal{H} \neq \varnothing$, compute A^{evo}_i using the linked skill utilities
  Optimize the masked joint objective to obtain $\pi_\theta^{(t+1)}$
end for
return $\pi_\theta^{(T)}, \mathcal{S}^{(T)}$

The within-branch residual sums to zero over g, so it cancels when credit is averaged for a selection decision. With population-form empirical standard deviations, the same structure yields

$$
\begin{aligned}
(\sigma_i^{\text{flat}})^2 &= (\sigma_i^{\text{sel}})^2 + \frac{1}{K}\sum_k \sigma_{i,k}^2, \\
(\sigma_i^{\text{sel}})^2 &= \frac{1}{K}\sum_k (\bar{r}_{i,k} - \mu_i)^2, \qquad \sigma_{i,k}^2 = \frac{1}{G}\sum_g (r_{i,k,g} - \bar{r}_{i,k})^2.
\end{aligned}
\tag{29}
$$

Normalizing each component by its own comparison group gives the main-text selection and utilization advantages:

$$
A^{\text{sel}}_{i,k} = \frac{\bar{r}_{i,k} - \mu_i}{\sigma_i^{\text{sel}} + \eta}, \qquad A^{\text{util}}_{i,k,g} = \frac{r_{i,k,g} - \bar{r}_{i,k}}{\sigma_{i,k} + \eta}. \tag{30}
$$

In particular, flat credit can be reconstructed as

$$
A^{\text{flat}}_{i,k,g} = \frac{\sigma_i^{\text{sel}} + \eta}{\sigma_i^{\text{flat}} + \eta} A^{\text{sel}}_{i,k} + \frac{\sigma_{i,k} + \eta}{\sigma_i^{\text{flat}} + \eta} A^{\text{util}}_{i,k,g}. \tag{31}
$$

Our method routes these components to different decisions and normalizes them separately, rather than broadcasting their sum. It is therefore a structured extension of GRPO, not an algebraically identical objective. Finite-group centering and scaling need not give an unbiased estimator of Eq. 27; a constant-reward group has zero advantage.

Why group-level evolution needs credit. One proposal $e_i = \tilde{s}_i$ is sampled after the complete outcome group $\mathcal{O}_i$ has been observed. Conditional on its history h_i, any source score $R(\mathcal{O}_i)$ is fixed. Consequently,

$$
\mathbb{E}_{e_i|h_i}[R(\mathcal{O}_i)\nabla_\theta \log \pi_\theta(e_i \mid h_i)] = R(\mathcal{O}_i)\nabla_\theta \sum_{e_i} \pi_\theta(e_i \mid h_i) = 0. \tag{32}
$$

Thus the observed group score cannot distinguish alternative reflections on that same group. A proposal instead receives credit after its cross-sample construction group produces a skill and that skill is reused. For skill j, the per-query EMA expands to

$$u_j^{(n)} = (1-\alpha)^n u_0 + \alpha \sum_{\ell=1}^{n} (1-\alpha)^{n-\ell} \bar{r}_j^{(\ell)}, \qquad \bar{r}_j^{(\ell)} = \frac{1}{G} \sum_g r_{j,\ell,g}. \tag{33}$$

The utility $u_{j(i)}$ is returned to every contributing proposal i. Normalizing $\hat{u}_i = u_{j(i)}$ across the feedback-bearing proposal batch $\mathcal{H}$ yields Eq. 4. Proposals linked to the same skill share a target, so this is group-level transfer credit, not an estimate of each proposal's isolated contribution. Unlike the within-query selection and utilization groups, $\mathcal{H}$ spans different construction groups. This utility term extends the task-reward objective.

Recovering the stage-masked objective. Let $\ell_\epsilon(\rho, A) = \min\{\rho A, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}$. For a stored decision record b at stage c, define

$$L_b^c(\theta) = \frac{1}{|\mathcal{M}_b^c|} \sum_{m \in \mathcal{M}_b^c} \ell_\epsilon\left(\frac{\pi_\theta(y_{b,m} \mid h_{b,m})}{\pi_{\mathrm{beh},b}(y_{b,m} \mid h_{b,m})}, A_b^c \right). \tag{34}$$

Here $\pi_{\mathrm{beh},b}$ is the policy that generated that record; its token probabilities are retained for updates. Rewards, utilities, and advantages are fixed targets during optimization. Selection records are indexed by (i,k), utilization records by (i,k,g), and evolution records by the task-group proposal index i. Their weighted averages give

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_i \left[\frac{1}{KG} \sum_{k,g} L_{i,k,g}^{\mathrm{util}} + \frac{\lambda_{\mathrm{sel}}}{K} \sum_k L_{i,k}^{\mathrm{sel}} \right] \\ + \lambda_{\mathrm{evo}} \mathbb{E}_{\mathcal{H}} \left[\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} L_i^{\mathrm{evo}} \right] - \beta D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}). \end{aligned} \tag{35}$$

This recovers Eq. 6 with the appropriate sampling units: one credit per selection, one per utilization trajectory, and one per task-group reflection. Each credit is applied only to its stage's tokens. The evolution term is omitted when no eligible feedback-bearing batch is available.

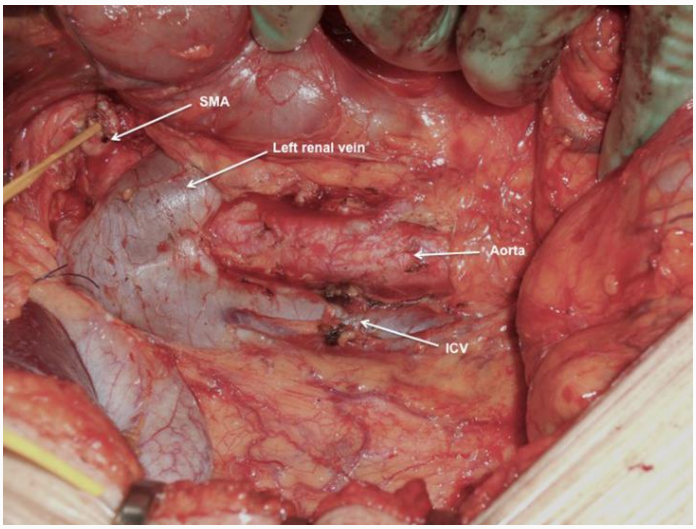
E CASE STUDIES

We present three qualitative case studies from the medical, mathematical, and chart–document domains to illustrate how retrieved skills improve multimodal reasoning. Each case provides the original image and question, the retrieved candidate skills and selected skill, and a comparison between the skill-assisted and base-model responses. These examples highlight how reusable skills help the model better translate perceived evidence into consistent decisions, appropriate geometric relations, and task-relevant reasoning operations.

E.1 CASE 1: MEDICAL ANATOMY

Case 1 | Medical anatomy

Original image



Complete question and reference answer

Where is the origin of the superior mesenteric artery identified?

Options:
A. immediately above the left renal vein
B. immediately below the left renal vein
C. approximately 5cm below the left renal vein
D. approximately 5cm above the left renal vein

Reference answer: A

Retrieval query

Identify the location of the origin of the superior mesenteric artery relative to the left renal vein.

Candidate 1/5: s_254fe316ed6216c2

{
"card": {
"skill_id": "s_254fe316ed6216c2",
"visual_domain": "medical",
"task_pattern": "identify",
"operation_type": "perception",
"description": "The Skill provides the capability to identify the correct anatomical structure or organ based on the provided information or visual representation, without being influenced by sample−specific entities, values, answer choices, or final answers. This involves locating and interpreting visual cues to determine the function or origin of a substance or process.",
"strategy": "1. Analyze the provided visual information or diagram. 2. Locate the relevant anatomical structure or organ based on the visual cues. 3. Use prior knowledge or the provided information to

Case 1 | Medical anatomy (continued)

determine the function or origin of the substance or process. 4. Identify the correct anatomical structure or organ that corresponds to the function or process.",
"utility": 0.73,
"num_uses": 9,
"active": true
},
"skill_id": "s_254fe316ed6216c2",
"similarity": 0.5372447371482849,
"utility": 0.73
}

Candidate 2/5: s_f598a60fda9d296a

{
"card": {
"skill_id": "s_f598a60fda9d296a",
"visual_domain": "medical",
"task_pattern": "identify",
"operation_type": "perception",
"description": "The Skill provides the capability to interpret visual cues in medical images, particularly focusing on the direction and location of arrows to identify specific anatomical structures or regions of interest. This involves analyzing the context and characteristics of the visual elements to infer their significance and function.",
"strategy": "1. Identify the direction and location of the arrow in the medical image. 2. Analyze the context and characteristics of the visual elements indicated by the arrow. 3. Correlate the visual cues with known anatomical structures or conditions based on the context and typical medical imaging findings. 4. Match the identified structure or region with the provided options or answer choices, if applicable.",
"utility": 0.61,
"num_uses": 6,
"active": true
},
"skill_id": "s_f598a60fda9d296a",
"similarity": 0.5117858052253723,
"utility": 0.61
}

Candidate 3/5: s_cf9dced55d97773e

{
"card": {
"skill_id": "s_cf9dced55d97773e",
"visual_domain": "medical",
"task_pattern": "identify",
"operation_type": "perception",
"description": "The Skill provides the capability to identify anatomical connections and structures in medical images and diagrams, focusing on the relationships between different parts of the body, such as the respiratory system. This behavior is applicable when analyzing medical images or diagrams to understand the flow of substances or the correct pathway between anatomical structures.",
"strategy": "1. Analyze the visual elements in a medical image or diagram to identify the anatomical connections between different parts of the body. 2. Locate and interpret labeled anatomical structures in a diagram to determine the correct connection or function. 3. Understand the function and position of a specific anatomical structure to identify it based on its location and role in the body. 4. Focus on the direct anatomical connections between different structures in a biological system based on visual cues and logical reasoning. 5. Exclude irrelevant options and focus on the correct anatomical structure that serves as a pathway between the upper and lower respiratory tracts.",
"utility": 0.58,
"num_uses": 8,
"active": true
},
"skill_id": "s_cf9dced55d97773e",
"similarity": 0.4987882077693939,
"utility": 0.58
}

Case 1 | Medical anatomy (continued)

Candidate 4/5: s_7015b874590183c7


```
{
  "card": {
    "skill_id": "s_7015b874590183c7",
    "visual_domain": "medical",
    "task_pattern": "identify",
    "operation_type": "perception",
    "description": "The Skill provides the capability to identify the correct anatomical structure in a
    diagram by observing visual cues, connections, and labels. This behavior is applicable when the task
    involves locating and determining the anatomical pathway or structure that connects two specific
    body parts or regions in a labeled diagram.",
    "strategy": "1. Analyze the diagram and identify the anatomical structures and labels provided. 2.
    Locate the anatomical pathway or structure that connects the two specific body parts or regions
    mentioned in the question. 3. Match the identified structure with the description or question to
    confirm the correct anatomical structure. 4. Verify the identified structure by checking its connections
     and labels in the diagram.",
    "utility": 0.56,
    "num_uses": 5,
    "active": true
  },
  "skill_id": "s_7015b874590183c7",
  "similarity": 0.49545377492904663,
  "utility": 0.56
}
```


Candidate 5/5: s_051b7f16b4c37ce4


```
{
  "card": {
    "skill_id": "s_051b7f16b4c37ce4",
    "visual_domain": "medical",
    "task_pattern": "identify",
    "operation_type": "perception",
    "description": "The Skill provides the capability to analyze medical images to identify specific
    anatomical structures or features indicated by visual markers such as arrows, and to interpret the
    context of the image to determine the significance of the indicated area or condition.",
    "strategy": "1. Identify the visual marker in the medical image, such as an arrow. 2. Analyze the
    context of the image, including the surrounding structures and the medical context. 3. Correlate the
    visual marker with the known anatomy of the region to identify the specific anatomical structure or
    feature being indicated. 4. Interpret the significance of the indicated area or condition based on the
    visual marker and the medical context.",
    "utility": 0.60,
    "num_uses": 7,
    "active": true
  },
  "skill_id": "s_051b7f16b4c37ce4",
  "similarity": 0.4928593635559082,
  "utility": 0.60
}
```


Reranking output (verbatim)


```
{"ranking":["s_254fe316ed6216c2","s_f598a60fda9d296a","s_cf9dced55d97773e","
   s_7015b874590183c7","s_051b7f16b4c37ce4"],"selected_skill_id":"s_254fe316ed6216c2"}
```


Selected skill (complete)

Selected skill:


```
{
  "skill_id": "s_254fe316ed6216c2",
  "visual_domain": "medical",
  "task_pattern": "identify",
  "operation_type": "perception",
```

Case 1 | Medical anatomy (continued)

"description": "The Skill provides the capability to identify the correct anatomical structure or organ based on the provided information or visual representation, without being influenced by sample−specific entities, values, answer choices, or final answers. This involves locating and interpreting visual cues to determine the function or origin of a substance or process.",
"strategy": "1. Analyze the provided visual information or diagram. 2. Locate the relevant anatomical structure or organ based on the visual cues. 3. Use prior knowledge or the provided information to determine the function or origin of the substance or process. 4. Identify the correct anatomical structure or organ that corresponds to the function or process.",
"utility": 0.73,
"num_uses": 9,
"active": true
}

Skill-assisted response: correct (complete)

<think>
The image shows the superior mesenteric artery (SMA) labeled and positioned above the left renal vein. The arrow points directly to the SMA, which is clearly marked and located just above the left renal vein. This aligns with the first option, which states the SMA is "immediately above the left renal vein."
</think>
<answer>
A. immediately above the left renal vein
</answer>

Base-model response: incorrect (complete)

<think>
The superior mesenteric artery (SMA) is shown in the image, and it is clearly labeled. The SMA is positioned above the left renal vein, which is also labeled. The SMA is not immediately above the left renal vein, but rather it is positioned above it, indicating that it originates from a point above the left renal vein.
</think>
<answer>B. immediately below the left renal vein</answer>

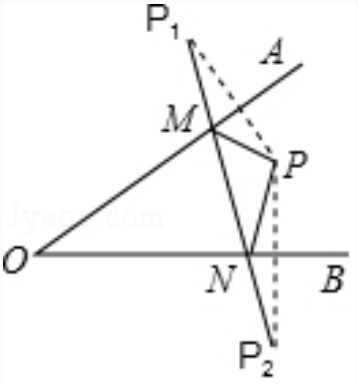
Reading the contrast

The contrast highlights the benefit of an explicit anatomical-localization skill. Although the base model recognizes that the SMA lies above the left renal vein, it fails to preserve this visual evidence when producing the final answer, resulting in a contradictory choice. In contrast, the selected skill structures the reasoning around locating the relevant anatomical structures, interpreting their spatial relationship, and grounding the decision in that relation. This leads to a consistent evidence-to-answer mapping and the correct choice, illustrating how reusable skills can stabilize multimodal reasoning beyond merely recognizing the right visual cues.

E.2 Case 2: Geometry

Case 2 | Geometry

Original image



Complete question and reference answer

Given: In the figure, point P inside ∠AOB, P~1~ and P~2~ are symmetric points of P with respect to OA and OB, respectively. P~1~P~2~ intersects OA at M and OB at N. If P~1~P~2~ = 6cm, what is the perimeter of triangle PMN?

Options:
A. 3cm
B. 4cm
C. 5cm
D. 6cm

Reference answer: D

Retrieval query

Identify reusable geometric reasoning strategies for problems involving reflection symmetry across the sides of an angle, equal−length relations induced by symmetric points, and using these relations to determine the perimeter of a triangle.

Candidate 1/5: s_92ffeedb49ae02d3


```
{
 "card": {
  "skill_id": "s_92ffeedb49ae02d3",
  "visual_domain": "mathematical",
  "task_pattern": "solve",
  "operation_type": "reasoning",
  "description": "The Skill involves recognizing and applying the properties of perpendicular bisectors to simplify geometric problems, particularly in calculating the perimeter of triangles. This includes identifying that the perpendicular bisector divides a segment into two equal parts and using this property to deduce equal segment lengths, which can be used to simplify the calculation of the perimeter.",
  "strategy": "1. Identify the presence of a perpendicular bisector in the geometric figure. 2. Recognize that the perpendicular bisector divides the segment it bisects into two equal parts. 3. Use the property of equal segment lengths to simplify the calculation of the perimeter of the triangle. 4. Apply the given perimeter or side lengths to find the unknown lengths and calculate the total perimeter.",
  "utility": 0.76,
  "num_uses": 10,
  "active": true
 },
 "skill_id": "s_92ffeedb49ae02d3",
 "similarity": 0.6579651236534119,
 "utility": 0.76
}
```

Case 2 | Geometry (continued)

Candidate 2/5: s_a99cee5ff9b7367f


```
{
  "card": {
    "skill_id": "s_a99cee5ff9b7367f",
    "visual_domain": "mathematical",
    "task_pattern": "solve",
    "operation_type": "reasoning",
    "description": "The Skill provides the capability to calculate the perimeter of a triangle by identifying
     the relevant geometric properties, applying the appropriate formula, and summing the lengths of the
    sides. This behavior is applicable when the task involves solving for the perimeter of a triangle,
    regardless of the specific type of triangle or the method used to find the missing side.",
    "strategy": "1. Identify the type of triangle (e.g., right, equilateral, isosceles). 2. Apply the appropriate
     geometric theorem or trigonometric function to find the missing side if necessary. 3. Sum the lengths
     of all sides to calculate the perimeter. 4. If rounding is required, round the result to the nearest tenth
    .",
    "utility": 0.62,
    "num_uses": 7,
    "active": true
  },
  "skill_id": "s_a99cee5ff9b7367f",
  "similarity": 0.6390674710273743,
  "utility": 0.62
}
```


Candidate 3/5: s_ad58a12f1efcc326


```
{
  "card": {
    "skill_id": "s_ad58a12f1efcc326",
    "visual_domain": "mathematical",
    "task_pattern": "solve",
    "operation_type": "reasoning",
    "description": "The Skill provides the capability to recognize and apply the properties of geometric
    shapes, particularly focusing on the equality of opposite sides in a parallelogram, to simplify
    calculations and determine the perimeter of the shape.",
    "strategy": "1. Identify the geometric shape in the problem. 2. Recognize if the shape is a
    parallelogram. 3. Apply the property that opposite sides of a parallelogram are equal in length. 4.
    Sum the lengths of all sides to find the perimeter. 5. Use the identified properties to simplify the
    calculation and solve the problem.",
    "utility": 0.55,
    "num_uses": 6,
    "active": true
  },
  "skill_id": "s_ad58a12f1efcc326",
  "similarity": 0.6006361246109009,
  "utility": 0.55
}
```


Candidate 4/5: s_5aeb5793ae8a10c5


```
{
  "card": {
    "skill_id": "s_5aeb5793ae8a10c5",
    "visual_domain": "mathematical",
    "task_pattern": "solve",
    "operation_type": "reasoning",
    "description": "The Skill provides a reusable visual reasoning capability that involves identifying and
    applying geometric properties and relationships to solve geometric problems. This includes
    recognizing and utilizing properties of triangles, parallelograms, and other geometric shapes, such as
    midlines, angle bisectors, and perpendiculars, to derive unknown lengths or solve for specific
    properties of the shapes.",
    "strategy": "1. Identify the geometric properties and relationships present in the problem, such as the
    properties of triangles, parallelograms, and other shapes. 2. Apply the relevant geometric principles,
```

Case 2 | Geometry (continued)

such as the properties of midlines, angle bisectors, and perpendiculars, to derive unknown lengths or solve for specific properties of the shapes. 3. Use the identified properties and relationships to simplify the problem and find the solution.",
"utility": 0.64,
"num_uses": 8,
"active": true
},
"skill_id": "s_5aeb5793ae8a10c5",
"similarity": 0.5655028820037842,
"utility": 0.64
}

Candidate 5/5: s_d2cea4e5bc7add58

{
"card": {
"skill_id": "s_d2cea4e5bc7add58",
"visual_domain": "mathematical",
"task_pattern": "solve",
"operation_type": "reasoning",
"description": "The Skill provides a capability to identify and utilize geometric properties of midlines, medians, and midsegments in various geometric shapes to deduce lengths of segments and calculate perimeters or other geometric properties. This is applicable when dealing with triangles, quadrilaterals, and parallelograms, where midlines, medians, and midsegments have specific properties that can be used to simplify calculations and find relationships between different segments of the shapes.",
"strategy": "1. Identify the geometric shape and the specific property of midlines, medians, or midsegments that is relevant to the problem. 2. Apply the identified property to deduce the length of a segment or calculate the perimeter of a shape. 3. Use the properties of midlines, medians, and midsegments to simplify the problem and find the required geometric properties.",
"utility": 0.60,
"num_uses": 5,
"active": true
},
"skill_id": "s_d2cea4e5bc7add58",
"similarity": 0.5638992190361023,
"utility": 0.60
}

Reranking output

{"ranking":["s_92ffeedb49ae02d3","s_a99cee5ff9b7367f","s_5aeb5793ae8a10c5","s_d2cea4e5bc7add58","s_ad58a12f1efcc326"],"selected_skill_id":"s_92ffeedb49ae02d3"}

Selected skill (complete)

Selected skill:

{
"skill_id": "s_92ffeedb49ae02d3",
"visual_domain": "mathematical",
"task_pattern": "solve",
"operation_type": "reasoning",
"description": "The Skill involves recognizing and applying the properties of perpendicular bisectors to simplify geometric problems, particularly in calculating the perimeter of triangles. This includes identifying that the perpendicular bisector divides a segment into two equal parts and using this property to deduce equal segment lengths, which can be used to simplify the calculation of the perimeter.",
"strategy": "1. Identify the presence of a perpendicular bisector in the geometric figure. 2. Recognize that the perpendicular bisector divides the segment it bisects into two equal parts. 3. Use the property of equal segment lengths to simplify the calculation of the perimeter of the triangle. 4. Apply the given perimeter or side lengths to find the unknown lengths and calculate the total perimeter.",
"utility": 0.76,
"num_uses": 10,
"active": true

Case 2 | Geometry (continued)

}

Skill-assisted response: correct (complete)

<think>
The problem involves a point P inside ∠AOB with symmetric points P~1~ and P~2~ with respect to OA and OB, respectively. The line P~1~P~2~ intersects OA at M and OB at N. The length of P~1~P~2~ is given as 6 cm. The task is to find the perimeter of triangle PMN.

Since P~1~ and P~2~ are symmetric with respect to OA and OB, the segments PM and PN are equal to the segments PM~1~ and PN~2~, respectively. This implies that the perimeter of triangle PMN is equal to the length of P~1~P~2~, which is 6 cm.
</think>
<answer>
D. 6cm
</answer>

Base-model response: incorrect (complete)

<think>
The problem involves a geometric configuration where point P is inside ∠AOB, and P~1~ and P~2~ are symmetric points of P with respect to OA and OB, respectively. The line segment P~1~P~2~ intersects OA at M and OB at N. The perimeter of triangle PMN is to be found given that P~1~P~2~ = 6cm.

Since P~1~ and P~2~ are symmetric points of P with respect to OA and OB, respectively, the segments PM and PM~1~, PN and PN~2~, and P~1~M and P~2~N are equal. This implies that triangle PMN is similar to triangle P~1~P~2~N and P~1~M~2~N, and the perimeter of triangle PMN is half the length of P~1~P~2~ because the segments are symmetrically divided.

Therefore, the perimeter of triangle PMN is 6cm / 2 = 3cm.
</think>
<answer>3cm</answer>

Reading the contrast

The contrast shows how the retrieved geometric skill turns reflection symmetry into the correct equal-length constraints. The base model recognizes the symmetric configuration but incorrectly interprets it as implying that the target perimeter should be half of P_1P_2, leading to 3 cm. With the skill, the model instead uses the reflection-induced relations $PM = P_1M$ and $PN = P_2N$, allowing the triangle perimeter to be rewritten along the collinear segment P_1P_2 and yielding the correct 6 cm. This illustrates how a reusable geometric skill can convert a recognized visual relation into the appropriate reasoning operation rather than an unsupported heuristic.

E.3 CASE 3: TABLE REASONING

Case 3 | Table reasoning

Original image

Destination	Elevation (ft.)
Underground lake	−350
Mountain peak	325

Complete question and reference answer

Manuel's campground is near two hiking trails with scenic endpoints. One trail leads to an underground lake at the bottom of a small cave. The other trail leads to a mountain peak. The elevation of each endpoint is shown on the campground map. Which trail's endpoint is farther from sea level?

Options:
A. underground lake
B. mountain peak

Reference answer: A

Retrieval query

Comparing the absolute values of the elevations to determine which is farther from sea level. The operation required is a comparison of the absolute values of the elevations.

Candidate 1/5: s_31bed783874c756f


```
{
  "card": {
    "skill_id": "s_31bed783874c756f",
    "visual_domain": "chart_document",
    "task_pattern": "compare",
    "operation_type": "perception",
    "description": "The Skill provides the capability to determine which of two locations is farther from sea level by comparing the absolute values of their elevations. This behavior is applicable when the task involves comparing the distances from sea level for two locations, endpoints, or destinations.",
    "strategy": "1. Identify the two locations, endpoints, or destinations to be compared. 2. Calculate the absolute values of their elevations. 3. Compare the absolute values to determine which one is farther from sea level. 4. The location with the greater absolute value is farther from sea level.",
    "utility": 0.74,
    "num_uses": 8,
    "active": true
  },
  "skill_id": "s_31bed783874c756f",
  "similarity": 0.7631354928016663,
  "utility": 0.74
}
```


Candidate 2/5: s_2788daa8385a93a9


```
{
  "card": {
    "skill_id": "s_2788daa8385a93a9",
    "visual_domain": "chart_document",
    "task_pattern": "compare",
```

Case 3 | Table reasoning (continued)


```
    "operation_type": "perception",
    "description": "The Skill provides the capability to compare the relative positions of data points or
    lines on a chart to determine which value or line is higher at a specific x−axis label. This behavior is
    applicable when the task involves comparing values or positions of data points or lines on a chart.",
    "strategy": "1. Identify the x−axis label in question. 2. Locate the data points or lines corresponding
    to the x−axis label. 3. Compare the y−axis values of the data points or lines at that point to determine
    which one is higher. 4. Apply this comparison to answer the question or solve the problem.",
    "utility": 0.57,
    "num_uses": 5,
    "active": true
  },
  "skill_id": "s_2788daa8385a93a9",
  "similarity": 0.5599378347396851,
  "utility": 0.57
}
```


Candidate 3/5: s_1e445ee496ffe54d


```
{
  "card": {
    "skill_id": "s_1e445ee496ffe54d",
    "visual_domain": "chart_document",
    "task_pattern": "compare",
    "operation_type": "perception",
    "description": "The Skill provides the capability to compare visual elements in a chart or graph to
    determine relative values. This involves identifying the positions, heights, or visual extent of elements
     to infer their relative magnitudes.",
    "strategy": "1. Identify the visual elements to be compared in the chart or graph. 2. Determine the
    relative positions, heights, or visual extent of these elements. 3. Use the identified relative positions,
    heights, or visual extent to infer the relative values of the elements.",
    "utility": 0.61,
    "num_uses": 7,
    "active": true
  },
  "skill_id": "s_1e445ee496ffe54d",
  "similarity": 0.5444610118865967,
  "utility": 0.61
}
```


Candidate 4/5: s_817ea85dbcdad0ae


```
{
  "card": {
    "skill_id": "s_817ea85dbcdad0ae",
    "visual_domain": "chart_document",
    "task_pattern": "compare",
    "operation_type": "perception",
    "description": "The Skill provides the capability to compare visual elements in a chart or document to
     determine relative magnitudes, such as the relative darkness of colors, brightness, sizes of sections,
    or positions along an axis. This behavior is applicable when the task involves comparing two entities
    within a visual representation to ascertain which one is greater or lighter/darker.",
    "strategy": "1. Identify the two entities to be compared within the visual representation. 2. Determine
    the visual attribute to use for comparison, such as color darkness, brightness, size, or position along
    an axis. 3. Compare the identified attributes to determine which entity is greater or lighter/darker. 4.
    Apply the comparison to answer the question or solve the problem.",
    "utility": 0.59,
    "num_uses": 6,
    "active": true
  },
  "skill_id": "s_817ea85dbcdad0ae",
  "similarity": 0.5436846017837524,
  "utility": 0.59
}
```

Case 3 | Table reasoning (continued)

Candidate 5/5: s_a9fb2c4e4c9b0fef


```
{
  "card": {
    "skill_id": "s_a9fb2c4e4c9b0fef",
    "visual_domain": "chart_document",
    "task_pattern": "compare",
    "operation_type": "perception",
    "description": "The Skill provides the capability to compare visual attributes (such as positions, sizes,
     and intensities) of entities (like lines, colors, or sections) in various types of charts and graphs to
    determine which one is greater or larger. This behavior is applicable when the task involves
    comparing two entities and requires identifying their relative positions, sizes, or intensities within the
    visual representation.",
    "strategy": "1. Identify the two entities to be compared in the chart or graph. 2. Observe the relative
    positions, sizes, or intensities of the entities. 3. Determine which entity is greater or larger based on
    the observed comparison. 4. Apply the comparison to the specific task at hand, such as identifying
    which line is greater on a graph, which color has a greater shade in a chart, or which section of a pie
    chart occupies a larger portion.",
    "utility": 0.57,
    "num_uses": 4,
    "active": true
  },
  "skill_id": "s_a9fb2c4e4c9b0fef",
  "similarity": 0.5420344471931458,
  "utility": 0.57
}
```


Reranking output


```
{"ranking":["s_31bed783874c756f","s_1e445ee496ffe54d","s_2788daa8385a93a9","
   s_817ea85dbcdad0ae","s_a9fb2c4e4c9b0fef"],"selected_skill_id":"s_31bed783874c756f"}
```


Selected skill (complete)

Selected skill:


```
{
  "skill_id": "s_31bed783874c756f",
  "visual_domain": "chart_document",
  "task_pattern": "compare",
  "operation_type": "perception",
  "description": "The Skill provides the capability to determine which of two locations is farther from
    sea level by comparing the absolute values of their elevations. This behavior is applicable when the
    task involves comparing the distances from sea level for two locations, endpoints, or destinations.",
  "strategy": "1. Identify the two locations, endpoints, or destinations to be compared. 2. Calculate the
    absolute values of their elevations. 3. Compare the absolute values to determine which one is farther
    from sea level. 4. The location with the greater absolute value is farther from sea level.",
  "utility": 0.74,
  "num_uses": 8,
  "active": true
}
```


Skill-assisted response: correct (complete)

<think>
The underground lake has an elevation of −350 feet, which means it is 350 feet below sea level. The mountain peak has an elevation of 325 feet, which means it is 325 feet above sea level. To determine which is farther from sea level, we compare the absolute values of these elevations. The absolute value of −350 is 350, and the absolute value of 325 is 325. Since 350 is greater than 325, the underground lake is farther from sea level.
</think>
<answer>
A. underground lake
</answer>

Case 3 | Table reasoning (continued)

Base-model response: incorrect (complete)

<think>
The underground lake is at an elevation of −350 feet, which means it is 350 feet below sea level. The mountain peak is at an elevation of 325 feet, which means it is 325 feet above sea level. Since the mountain peak is above sea level and the underground lake is below sea level, the mountain peak is farther from sea level.
</think>
<answer>B. mountain peak</answer>

Reading the contrast

The contrast directly shows the benefit of the retrieved comparison skill. Both responses correctly identify the elevations as -350 ft and 325 ft, but the base model incorrectly treats positive elevation as being farther from sea level. The selected skill instead specifies the appropriate operation: compare the absolute distances from the reference level. Following this strategy, the skill-assisted model explicitly computes $350 > 325$ and correctly selects the underground lake. This case illustrates how a reusable skill can supply the missing reasoning operation even when the underlying visual quantities have already been correctly perceived.

F Limitations and Future Work

Our evaluation focuses on multimodal reasoning tasks with verifiable answers and does not systematically distinguish failures of reasoning strategy from perceptual errors or missing world knowledge. Skill transfer primarily targets cases where the model can access the necessary evidence and knowledge but struggles to select or execute an appropriate reasoning strategy. It cannot be assumed to recover visual information the model cannot perceive or supply knowledge absent from both the policy and the skill library. Evaluating recovery separately across these failure modes and extending verification beyond answer correctness remain important directions for future work.

The fixed visual-state taxonomy and exact-match filtering may restrict reuse across categories and limit skill composition for tasks requiring multiple reasoning operations. Our evaluation also covers only a finite number of self-improvement rounds, leaving long-term stability and scalability untested. As the library grows, retrieval costs, redundant skills, and accumulated errors may affect continued improvement. Adaptive skill organization, explicit skill composition, and longer-horizon evaluations are needed to assess these risks. Extending the framework to open-ended tasks will additionally require reliable feedback beyond the verifiable-answer setting studied here.