AGENT REASONING WITH INTRINSIC SKILL EVOLUTION IN HIERARCHICAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards has become the dominant paradigm for mathematical reasoning in language models, yet most methods treat each problem as a fresh episode and retain discovered strategies only implicitly through policy updates. Abstracting reusable techniques and recognizing when to apply them, the central mechanism of mathematical expertise, therefore stays outside training. Unlike the executable procedures stored in skill libraries for tool-use and embodied agents, a mathematical skill is a reasoning pattern embedded in a solution trace, whose applicability must be inferred from the problem and whose worth is defined only against the same policy solving it unaided. Abstracting and selecting such skills are therefore reasoning tasks that existing designs leave outside policy optimization. We introduce ARISE (Agent Reasoning with Intrinsic Skill Evolution), a hierarchical reinforcement learning framework in which one shared policy serves as the high-level option policy that selects from a persistent skill library and as the low-level intra-option policy that generates solutions conditioned on the selected skill. The policy scores library entries through its own conditional log-likelihood and concludes new skills from ground-truth traces through an inference step. With one option per query, the option-critic gradient reduces to two terms under one group-relative advantage, and generation feedback reaches selection through the tangent kernel between skill documents and rollouts, so a single policy gradient optimizes both levels. The three-level trajectory reward, which separates skill-aided from unaided correct solutions, enters the advantage, while a counterfactual skill quality level governs library maintenance, so reasoning ability and library quality co-evolve. Across two instruction-tuned base models and seven competition and Olympiad-level benchmarks, ARISE outperforms GRPO-family algorithms and memory- and skill-augmented baselines, with the largest gains on out-of-distribution tasks, and the gains persist on a third backbone at 8B scale and in code generation. The implementation is available at ARISE Codebase.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for mathematical reasoning with large language models, allowing policies to self-improve from outcome feedback alone (Guo et al., 2025). Yet many existing methods treat each problem as a fresh episode: once a rollout terminates, the strategies discovered during reasoning, such as telescoping for series or auxiliary-line construction in geometry, are retained only implicitly through policy updates and unavailable for explicit reuse (Sun et al., 2025; Zhang et al., 2025). This limits a central mechanism of building mathematical expertise: by abstracting reusable techniques from individual solutions, organizing them into a repertoire of named strategies, and learning to recognize when and how to apply them to new problems.

Learning reusable skill libraries across episodes offers a natural way to address this limitation (Wang et al., 2025; Wei et al., 2026; Wu et al., 2025; Xia et al., 2026). However, existing approaches largely focus on tool-use and embodied agents, whose skills are often executable procedures and also admit clear applicability conditions grounded in observable task states and assessments through environmental feedback. Mathematical skills differ on both counts: a skill is an abstract reasoning pattern embedded in a solution trace, while its applicability must be inferred from the problem and assessed through counterfactual comparison with the same policy solving the problem unaided.

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.

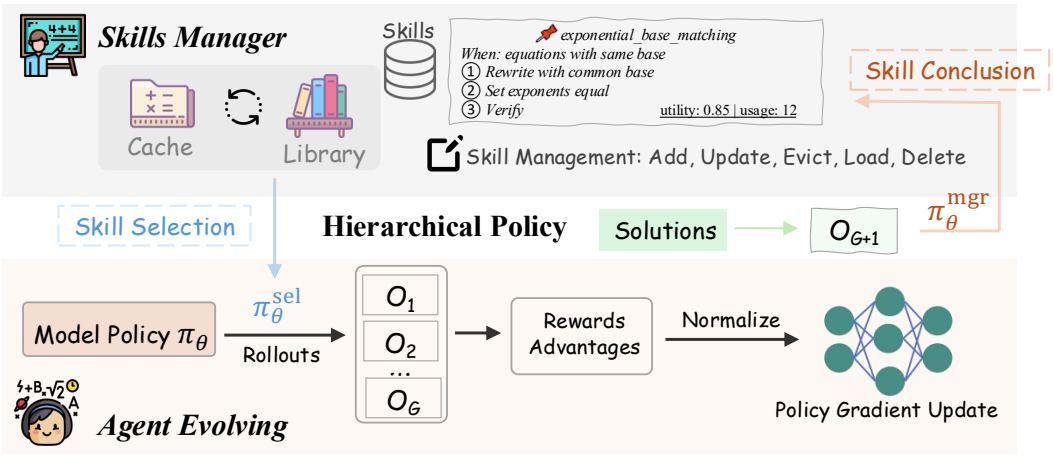


Figure 1: The figure gives an overview of ARISE, in which the shared policy π_θ acts as the policy over options that scores and selects a skill from the library (upper panel) and as the intra-option policy that generates solutions conditioned on the selected skill (lower panel). After each rollout group, the same policy performs skill conclusion through an inference-only step, and maintenance rules manage the library.

Abstracting a mathematical skill from a solution and selecting an appropriate skill for a new problem are complementary yet coupled reasoning tasks themselves. Yet existing designs often leave both reasoning tasks outside policy optimization and maintain the skill library through heuristics decoupled from the evolving policy (Hao et al., 2024). As a result, the library remains an external resource rather than an integral component of a unified learning process in which skill acquisition, selection, and application improve together.

To bridge this gap, we present ARISE, a hierarchical policy framework built around an Evolving-Skill MDP that treats the skill library as an endogenous component of the agent's state and each library entry as an option. One shared policy π_θ (through its factorization) occupies both levels of the hierarchy: as the high-level option policy, it scores every candidate by conditional log-probability and samples from a temperature-scaled distribution, and as the low-level intra-option policy, it generates solutions conditioned on the selected skill. Since both levels run on shared model parameter θ, one group-relative advantage shapes what skills are selected as well as how they are leveraged. With one option per query that ends with the sequence, the termination term of the option-critic gradient vanishes and the policy gradient reduces to two terms, the selection score and the solution log-probability, under one group-relative advantage. Since the score is the conditional log-likelihood of the skill document under π_θ, a gradient step on the solutions moves every score by the advantage-weighted inner product of its gradient with the rollout gradients, the tangent kernel through which generation feedback updates selection.

This approach enables a unified policy gradient avoiding the use of separate value functions or potential inconsistency of isolated architectures. After each rollout group, the same policy distills the ground-truth trace into a structured skill document via an inference-only rollout, while maintenance rules move entries between an active cache and a reservoir archive. The policy itself performs the abstraction, from the query and the ground-truth trace into a document with a fixed schema of name, problem type, key insight, method, and check, so a skill is a reasoning pattern in the policy's own distribution, which makes its conditional log-likelihood a selection score and its counterfactual verification a measure of worth. The three-level trajectory reward, which grades an incorrect solution ζ_0, a correct unaided solution ζ_1, and a correct solution under a selected skill ζ_2, enters the group advantage, while the skill quality level R^s, measured by a counterfactual verification rollout, governs library maintenance. Within a group of G rollouts on one query, the advantage of a rollout is its trajectory reward centered by the group mean and scaled by the group standard deviation, and it multiplies both the selection score and the solution log-probability. Training proceeds in two phases, a warm-up that builds base competence before the policy over options activates and a skill-augmented phase in which policy and library co-evolve.

We instantiate and validate ARISE on mathematical reasoning with Qwen3-4B-Instruct-2507 (Team, 2025) and Phi-4-mini-instruct (Abouelenin et al., 2025), trained on DeepScaleR and tested on AMC 2023, AIME 2024, AIME 2025, and the four domains of the out-of-distribution Olympiad benchmark Omni-MATH (Gao et al., 2024). On both base models, ARISE improves over GRPO-family baselines and memory- and skill-augmented methods, with the largest gains out of distribution and the gains of the three-seed Qwen3-4B configurations exceeding the seed standard deviation, and the gains persist on Qwen3-8B-Base and carry over to code generation.

❶ We formalize the Evolving-Skill MDP and a hierarchical policy factorization in which one shared policy serves as the high-level option policy and as the low-level intra-option policy, unifying skill selection and solution generation. We derive the option-critic reduction and the tangent-kernel channel through which generation feedback updates selection. ❷ We design a self-evolving skill cycle in which the policy concludes skills through inference-time abstraction and a counterfactual comparison grades their marginal worth, driving co-evolution between policy and library free of auxiliary signals or networks. ❸ Across two base models and seven benchmarks, ARISE outperforms GRPO-family and skill-augmented baselines under iso-compute and three-seed statistics, ablations attribute the gains to the shared-parameter skill mechanism, and the gains transfer to a third backbone and to code generation.

2 Preliminaries

2.1 Related Work

Memory- and skill-augmented agents. External memory and reusable skills form one complementary line to RLVR for LLM agents (Zhou et al., 2026). Inference-time methods such as Reflexion (Shinn et al., 2023), ExpeL (Zhao et al., 2024), and SimpleMem (Liu et al., 2026) retrieve past trajectories or distilled insights into context, and Voyager (Wang et al., 2023) and DynaSaur (Nguyen et al., 2024) grow libraries of executable skills for embodied and tool-use agents, with the memory populated and retrieved independently of policy learning in each case. Training-integrated methods embed skill structures into the RL loop: EvolveR (Wu et al., 2025) maintains experience narratives across GRPO training, SkillRL (Xia et al., 2026) targets embodied and search agents through a recursive multi-turn loop, and SAGE (Wang et al., 2025) accumulates skills across a task chain on AppWorld with a skill-integrated reward. In these three methods, the operations that decide which skill to invoke and which to retain sit outside the policy gradient.

Hierarchical reinforcement learning. The options framework (Sutton, 1998) extends MDPs with temporally extended actions, the option-critic architecture (Bacon et al., 2017) unifies the policy gradient across levels, and feudal designs (Vezhnevets et al., 2017) maintain separate manager and worker networks with decomposed rewards. Option discovery has been pursued through bottleneck states (Stolle & Precup, 2002), through goal-conditioned lower levels that follow subgoals set by a higher-level policy (Nachum et al., 2018), and through unsupervised objectives that reward distinguishable skills (Eysenbach et al., 2019), while Skill-critic (Hao et al., 2024) refines pre-learned skills jointly with the high-level policy. In LLM agents the higher level is prompting or retrieval, as in Voyager, DynaSaur, and SkillRL above, and a policy over options trained together with the lower level is absent.

2.2 Problem Setup

Under RLVR (Xu et al., 2025; Patil & Jadon, 2025; Wen et al., 2025; Alam & Rastogi, 2025; Li et al., 2026), let $\mathcal{Q}$ be the query space and $\mathcal{T}$ the trajectory space, with each $q \in \mathcal{Q}$ associated with a ground-truth answer $a^\star(q)$. The policy $\pi_\theta(\tau \mid q)$ generates a trajectory τ given q, and a verifier $v(\tau, q) \in \{0, 1\}$ checks whether the extracted answer matches $a^\star(q)$, defining the binary outcome reward of RLVR. Group Relative Policy Optimization (GRPO) (Guo et al., 2025) is a critic-free variant of PPO (Schulman et al., 2017): for each query it draws $\{\tau_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid q)$, forms group-relative advantages by normalizing the rewards with the group mean and standard deviation, and optimizes a clipped surrogate over per-token importance ratios. The group statistics absorb the role of a value baseline, §3.2 states the advantage and the surrogate in the skill-conditioned form, and the GRPO variants of §4 keep the memoryless per-query structure.

The hierarchical extension of §3 rests on the options formalism (Sutton, 1998). Each option $o = (\mathcal{I}, \pi^o, \beta)$ comprises an initiation set $\mathcal{I} \subseteq \mathcal{S}$, an intra-option policy π^o, and a termination function

$\beta : \mathcal{S} \to [0, 1]$. Over a finite option set Ω, the policy over options μ selects $o \in \Omega$ whenever the previous option terminates, and the chosen π^o generates actions until β fires. §3 instantiates the formalism over an augmented state pairing the query with the current skill library, treats each library entry as an option induced by π_θ itself, and unifies μ and $\{\pi^o\}$ under one set of shared parameters θ.

3 ARISE: Hierarchical Policy with Evolving Skills

ARISE is a hierarchical policy in which one shared model π_θ occupies both levels: as the policy over options it scores candidate skills from a persistent library and selects one for each query, and as the intra-option policy it generates solutions conditioned on the selected skill. Both levels share θ and are optimized through one policy gradient, while skill conclusion and library maintenance operate as gradient-free updates to the library state.

3.1 Evolving-Skill MDP and Hierarchical Policy

Evolving-Skill MDP. With $\mathcal{Q}$ and $\mathcal{T}$ as in §2.2 and $\mathcal{L}$ the set of finite-capacity libraries, the Evolving-Skill MDP is the tuple

$$\mathcal{M} = \big(\mathcal{S},\, \Omega,\, \mathcal{T},\, P,\, R^w,\, R^s\big), \qquad \mathcal{S} = \mathcal{Q} \times \mathcal{L}, \qquad \Omega(C) = \big\{o_k\big\}_{k=1}^{|C|} \cup \{o_\varnothing\}, \tag{1}$$

where a library $L = C \cup R$ comprises an active cache C, whose entries induce the options, and a reservoir R that archives displaced entries, each entry $m_k \in C$ induces an option $o_k = (I_k, \pi_{\theta,k}, \beta_k)$, and $o_\varnothing$ is the unaided fallback. The initiation set $I_k = \mathcal{S}$ makes the options applicable subject to the confidence gate below, the intra-option policy $\pi_{\theta,k}(\tau \mid q) = \pi_\theta(\tau \mid q, m_k)$ conditions the shared policy on the document m_k, and the termination β_k fires at the end-of-sequence token, so each option spans one full solution trajectory. The option set $\Omega(C_t)$ is time-varying, with the library evolving under the transition P realized as

$$L_{t+1} \sim K\big(\cdot \mid L_t,\, \tilde{m}_t,\, z_t,\, R^s\big), \qquad q_{t+1} \sim \mathcal{D}, \qquad \tilde{m}_t \sim \pi_\theta\big(\cdot \mid q_t,\, \tau_t^+\big), \tag{2}$$

where $\tilde{m}_t$ is the skill concluded from the ground-truth trace τ_t^+ by an inference-only rollout and K is the library kernel realized by the rules of §3.2. The coupling yields a non-stationary MDP whose option set is endogenous to the policy.

Hierarchical policy. Over the augmented state $s_t = (q_t, L_t)$, the policy over options μ_θ selects $o_{z_t} \in \Omega(C_t)$, and the selected intra-option policy π_{θ,z_t} generates the solution trace until β_{z_t} fires. Each query therefore invokes exactly one option that terminates at sequence end, and the joint distribution factorizes as

$$\pi_\theta\big(\tau_t, z_t \mid s_t\big) \;=\; \mu_\theta\big(z_t \mid s_t\big) \cdot \pi_{\theta,z_t}\big(\tau_t \mid q_t\big), \tag{3}$$

where exposing only the selected document to the intra-option policy enforces conditional independence of τ_t from L_t given m_{z_t}. μ_θ reuses π_θ in place of a separate scoring head: each option is scored by the length-normalized conditional log-likelihood of its document, and the scores drive a temperature softmax with a confidence gate,

$$\ell_\theta(o_k \mid s_t) \;=\; \frac{1}{|m_k|} \sum_{l=1}^{|m_k|} \log \pi_\theta\big(m_k^{(l)} \mid q_t, m_k^{(<l)}\big), \qquad \mu_\theta(o_k \mid s_t) \;=\; \frac{\exp\big(\ell_\theta(o_k \mid s_t)/\sigma\big)}{\sum_{j=1}^{|C_t|} \exp\big(\ell_\theta(o_j \mid s_t)/\sigma\big)}, \tag{4}$$

with the full probability mass moved to the fallback, $\mu_\theta(o_\varnothing \mid s_t) = 1$, whenever $\max_k \mu_\theta(o_k \mid s_t) < \delta$, so the unaided fallback is itself a valid branch of the policy over options. The temperature $\sigma > 0$ controls selection sharpness and $\delta \in (0, 1)$ guards against low-confidence selections. Since the scores come from π_θ itself, the policy over options is critic-free, and Appendix B.2 reads Eq. (4) as a length-corrected tempered Bayes posterior over cache entries under a uniform prior.

Unified policy gradient. Since μ_θ and $\{\pi_{\theta,k}\}$ are induced from one θ, the objective $J(\theta) = \mathrm{E}_{s_t, z_t, \tau_t \sim \pi_\theta}\big[\, R^w(\tau_t) \,\big]$ admits a single-parameter gradient, and the regime of §3.1 collapses the option-critic decomposition (Bacon et al., 2017) to the two-term update of Proposition 3.1.

Table 1: The table lists the library operations executed in fixed order after each rollout group, where $\bar{R}_t^w$ is the mean trajectory reward of the group.

OPERATION	EFFECT
UPDATE	$\bar{r}_{z_t} \leftarrow \lambda\, \bar{r}_{z_t} + (1-\lambda)\, \bar{R}_t^w$
ADD	Insert $\tilde{m}_t$ into C_t with $\bar{r}_{\tilde{m}_t} = R^s$
EVICT	Lowest-utility cache entry $\rightarrow R_t$
LOAD	Promote R_t entry if $\bar{r}_k$ exceeds weakest cache
DELETE	Remove R_t entry if $\bar{r}_k < u_{10}(R_t)$ and $n_k = 0$

Table 2: The table assigns the skill quality level from the verification outcome against the G original rollouts.

G rollouts	τ_{G+1}	R^s
all incorrect	correct	r_4
$\exists$ correct	correct	r_3
$\exists$ correct	incorrect	r_1
all incorrect	incorrect	r_2

Proposition 3.1 (Gradient reduction)**.** *In the Evolving-Skill MDP with call-and-return execution, one option per query, and termination fixed to the end-of-sequence indicator, the termination term of the option-critic gradient is identically zero and the policy gradient reduces to*

$$\nabla_\theta J(\theta) \;=\; \mathrm{E}\Big[\, \hat{A}_t\, \nabla_\theta \log \mu_\theta(z_t \mid s_t) \;+\; \hat{A}_t\, \nabla_\theta \log \pi_{\theta,z_t}(\tau_t \mid q_t) \,\Big], \tag{5}$$

with both terms carrying the group-relative advantage of Eq. (7) (proof in Appendix B.1).

The score gradient $h_k = \nabla_\theta\, \ell_\theta(o_k \mid s_t)$ and the rollout gradients $g_i = \nabla_\theta \frac{1}{|\tau_i|} \log \pi_\theta(\tau_i \mid q_t, m_{z_t})$ occupy one parameter space, and Proposition 3.2 tracks the selection scores through one update.

Proposition 3.2 (Selection channel)**.** *With one option shared by the G rollouts, the group Monte Carlo estimate of the selection term in Eq. (5) equals zero for every realized group. After one ascent step of size η on Eq. (8), the score of every cache entry satisfies*

$$\ell_{\theta^+}(o_k \mid s_t) - \ell_\theta(o_k \mid s_t) \;=\; \frac{\eta}{G} \sum_{i=1}^{G} \hat{A}_i \,\big\langle h_k,\, g_i \big\rangle + O(\eta^2), \tag{6}$$

so selection moves by the advantage-weighted tangent-kernel alignment (Jacot et al., 2018) between each document and the rollouts of the group (proof in Appendix B.1).

The surrogate of §3.2 therefore omits the selection term at zero estimation cost, and the update to μ_θ arrives only through the kernel channel of Eq. (6). The decoupled scoring head of §4.3 severs the channel and receives zero gradient from the group-relative advantage (Corollary B.1).

3.2 SELF-EVOLVING SKILL CYCLE

Selection and conditioned rollouts. Given q_t, μ_θ scores the entries of the active cache C_t through Eq. (4) and samples one option per query, falling back to $o_\varnothing$ whenever the highest candidate score stays below the confidence threshold δ. The G rollouts therefore share the selection, and the group advantage attributes outcome differences within the group to the conditioned generation. Upon a non-fallback selection, the skill document m_{z_t} is prepended to q_t as context; each document carries five structured fields, namely skill name, problem type, key insight, step-by-step method, and verification check, with an example in Appendix D.1. The selected intra-option policy π_{θ,z_t} then generates $\{\tau_i\}_{i=1}^G$, each graded by the trajectory reward $R_i^w \in \{\zeta_0, \zeta_1, \zeta_2\}$ of §3.3. Advantages follow the GRPO normalization

$$\hat{A}_i = \frac{R_i^w - \bar{R}_t^w}{s_G + \epsilon}, \qquad \bar{R}_t^w = \tfrac{1}{G}\textstyle\sum_{i=1}^G R_i^w, \qquad s_G^2 = \tfrac{1}{G}\textstyle\sum_{i=1}^G (R_i^w - \bar{R}_t^w)^2, \tag{7}$$

and drive the clipped surrogate

$$\mathcal{L}_G(\theta) = \frac{1}{G}\sum_{i=1}^{G} \frac{1}{|\tau_i|} \sum_{l=1}^{|\tau_i|} \min\Big(\rho_{i,l}\, \hat{A}_i,\; \mathrm{clip}\big(\rho_{i,l},\, 1-\epsilon_c,\, 1+\epsilon_c\big)\, \hat{A}_i\Big), \qquad \rho_{i,l} = \frac{\pi_\theta(x_{i,l} \mid c_{i,<l})}{\pi_{\theta'}(x_{i,l} \mid c_{i,<l})}, \tag{8}$$

where $x_{i,l}$ is the l-th token of τ_i, θ' denotes the parameters before the update, ϵ_c is the clipping width, and the context $c_{i,<l} = (q_t, m_{z_t}, x_{i,<l})$ reduces to $(q_t, x_{i,<l})$ under the unaided branch.

Conclusion and maintenance. After each group, the policy concludes $\tilde{m}_t \sim \pi_\theta(\cdot \mid q_t, \tau_t^+)$ through the inference step of Eq. (2). The verification rollout τ_{G+1} then solves q_t conditioned on $\tilde{m}_t$, and its

outcome against the G original rollouts yields the skill quality level R^s of §3.3, which initializes the running utility $\bar{r}_{\tilde{m}_t}$ of the entry. Each entry of $L_t = C_t \cup R_t$ carries its utility $\bar{r}_k$ and selection count n_k. The five operations of Table 1 execute in fixed order, with hyperparameters listed in Appendix C.1. UPDATE refreshes the selected entry with the mean trajectory reward $\bar{R}_t^w$ of Eq. (7) for the group that invoked it, so the initial value set by R^s is discounted by λ at later selections. EVICT and LOAD jointly realize a promotion-demotion mechanism between the two tiers: the lowest-utility entry moves to the reservoir whenever the cache exceeds its capacity, while reservoir entries that accumulate evidence of usefulness re-enter the active pool. DELETE garbage-collects reservoir entries with zero selections whose utility has decayed below the tenth-percentile threshold $u_{10}(R_t)$, preventing the reservoir from accumulating stale content.

3.3 REWARD WITH MARGINAL SKILL ATTRIBUTION

Of the two reward quantities in ARISE, only the trajectory reward enters the policy gradient. The trajectory reward

$$R_i^w = \begin{cases} \zeta_2, & v(\tau_i, q_t) = 1, \quad z_t \neq \varnothing, \\ \zeta_1, & v(\tau_i, q_t) = 1, \quad z_t = \varnothing, \\ \zeta_0, & v(\tau_i, q_t) = 0, \end{cases} \tag{9}$$

with $\zeta_0 < \zeta_1 < \zeta_2$ and the values of Appendix C.1, withholds reward from incorrect solutions regardless of skill use, so the policy is not reinforced for applying an irrelevant or misleading skill, and the third level becomes reachable only once the policy over options activates.

The skill quality level R^s measures the marginal contribution of each concluded skill through a counterfactual comparison: τ_{G+1} solves the same q_t conditioned on $\tilde{m}_t$, and its outcome against the G original rollouts yields one of the four levels $r_1 < r_2 < r_3 < r_4$ of Table 2. Level r_4 identifies skills with critical marginal contribution, since the skill enables solving a problem that the policy failed on in its G original attempts. r_3 recognizes effective but non-essential skills on already-solvable problems, r_1 penalizes misleading skills that degrade performance on otherwise-solvable problems, and r_2 provides a neutral baseline when neither the base policy nor the skill succeeds. The ordering reflects the partial order on counterfactual utility induced by the two binary outcomes, and the numerical values used in our experiments are listed in Appendix C.1. The comparison asks whether conditioning on $\tilde{m}_t$ changes the outcome relative to rollouts drawn for the same query, a contrast defined only when aided and original rollouts share the query, which separates it from a self-consistency check on the source query. The level initializes $\bar{r}_{\tilde{m}_t}$ and thereafter influences the entry only through the UPDATE rule, which refreshes the utility with the mean trajectory reward of the later groups that select it. The presence of an entry in the cache is therefore governed by its performance on later queries, not by its point of origin. For an entry with zero selections the source-query level remains the value that governs EVICT, and the cache capacity and the confidence gate bound the effect of an inflated initial value (Appendix E.2, Proposition B.1). Appendix H.1 recomputes R^s for the converged cache on unseen queries of the same problem type and finds positive attribution for eight of the ten entries.

3.4 TWO-PHASE TRAINING

Training proceeds in two phases, since the untrained policy cannot yet rank skills and a library populated by skills concluded by a weak policy would inject noise, with the skill cycle operating in both phases and the policy over options activated only in the second.

Phase I: Warm-up. During the first N_w steps, the policy over options is disabled and π_θ trains as a flat policy on the verifier outcome through Eq. (8). Skill conclusion executes once per training step, each entry is tested by a verification rollout whose outcome initializes its utility, and maintenance manages the cache-reservoir partition as entries accumulate. By the end of warm-up, π_θ has developed competent base reasoning and the library contains a diverse repertoire with calibrated utilities, the two preconditions of option selection.

Phase II: Skill-augmented training. Once warm-up concludes at step $N_w + 1$, the full hierarchical policy of Eq. (3) takes effect, with N_w set by the sensitivity study of Appendix F.1. In Phase II the group advantage reaches both levels through the shared parameters, and R^s continues to initialize the utility of each concluded entry. The conclusion source stays fixed to τ_t^+, and the quality of concluded entries improves as π_θ learns to extract more structured patterns from the same traces. Figure 2(b)

Algorithm 1 ARISE

Require: $\mathcal{D}$, π_{θ_0}, group size G, warm-up steps N_w, threshold δ

$C \leftarrow \emptyset$, $R \leftarrow \emptyset$
Phase I: Warm-up
for $t = 1, \ldots, N_w$ **do**
  for each $(q, \tau^+) \sim \mathcal{D}$ **do**
    $\{\tau_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid q)$; compute R_i^w, $\hat{A}_i$
    $\tilde{m} \sim \pi_\theta(\cdot \mid q, \tau^+)$ inference only
    Verify τ_{G+1}; $\bar{r}_{\tilde{m}} \leftarrow R^s$
    Add $\tilde{m}$ to C; run maintenance
  end for
  Update θ via $\mathcal{L}_G(\theta)$ Eq. (8)
end for
Phase II: Skill-Augmented
for $t = N_w + 1, N_w + 2, \ldots$ **do**
  for each $(q, \tau^+) \sim \mathcal{D}$ **do**
    Score options via Eq. (4); sample z_t if $\max_k \mu_\theta(o_k \mid s_t) \geq \delta$
    $\{\tau_i\}_{i=1}^{G} \sim \pi_{\theta, z_t}(\cdot \mid q)$; compute R_i^w, $\hat{A}_i$
    $\tilde{m} \sim \pi_\theta(\cdot \mid q, \tau^+)$ inference only
    Verify τ_{G+1}; $\bar{r}_{\tilde{m}} \leftarrow R^s$
    Update $\bar{r}_{z_t}$; add $\tilde{m}$; run maintenance
  end for
  Update θ via $\mathcal{L}_G(\theta)$ Eq. (8)
end for
return θ, C, R

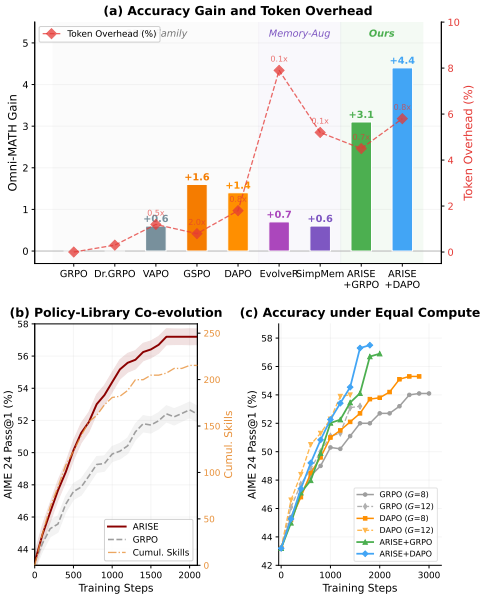


Figure 2: Panel (a) plots accuracy gain against token overhead across methods, panel (b) tracks policy-library co-evolution on AIME 2024 during training, and panel (c) compares accuracy trajectories under an equal compute budget.

shows the cumulative skill count and AIME 2024 accuracy rising together during Phase II, with the count plateauing once the cache settles into a stable repertoire.

4 Results

4.1 Implementation Details

The methods are trained on DeepScaleR (Luo et al., 2025b), approximately 40K problem-solution pairs from AMC, AIME, MATH, and OlympiadBench, with Qwen3-4B-Instruct-2507 (Team, 2025) and Phi-4-mini-instruct (Abouelenin et al., 2025), group size $G = 8$, and 2,000 training steps, with the compute-matched comparison in Figure 2(c). Baselines consume only the final answer of each pair, whereas ARISE also reads the reference solution during skill conclusion, and ablation (e) of §4.3 isolates the gain that remains when the traces are withheld. Each ARISE row is paired with the baseline sharing its optimizer: ARISE+GRPO uses the plain GRPO objective with dynamic sampling disabled and is compared with GRPO, and ARISE+DAPO inherits the dynamic sampling and asymmetric clipping of DAPO and is compared with DAPO. Baselines cover the GRPO family, namely GRPO (Guo et al., 2025), Dr.GRPO (Liu et al., 2025), VAPO (Yue et al., 2025), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025), and memory- and skill-augmented methods, namely EvolveR (Wu et al., 2025) and SimpleMem (Liu et al., 2026), each integrated with GRPO and adapted to our setting. SAGE (Wang et al., 2025) is built for multi-turn agents, so an adapted single-turn variant is reported separately in Appendix G.1. Evaluation covers the in-distribution competition benchmarks AMC 2023[1], AIME 2024, and AIME 2025[2], which share problem types with the training set while remaining temporally disjoint from it. Out-of-distribution evaluation uses Omni-MATH (Gao et al., 2024) with 4,428 Olympiad problems partitioned into Algebra, Number Theory, Combinatorics, and Geometry. Results report Pass@1 averaged over 32 evaluation samples under fixed weights, and the Qwen3-4B configurations marked with a deviation in Table 3 were retrained with three training seeds, for which the table gives the mean and the standard deviation across seeds.

[1] https://huggingface.co/datasets/AI-MO/aimo-validation-amc
[2] https://huggingface.co/datasets/AI-MO/aimo-validation-aime

Table 3: The table reports Pass@1 (%) on mathematical reasoning benchmarks, averaged over 32 evaluation samples with $G = 8$ on DeepScaleR. Gray deviations give the standard deviation over three training seeds where retrained. **Bold** marks the best value in each column within a base model. Green and blue rows denote our method, with subscripts giving the gain over GRPO under the same base model.

	In-distribution *competition*			**Out-of-distribution** *Omni-MATH*				
Method	AMC 23	AIME 24	AIME 25	Algebra	Num.Thy	Combin.	Geometry	Avg.†
Qwen3-4B-Instruct-2507								
Base model	69.6	53.6	45.6	30.1	21.2	13.0	20.5	21.2
GRPO family								
GRPO (Guo et al., 2025)	72.9$_{\pm 0.4}$	54.1$_{\pm 0.7}$	46.5$_{\pm 0.8}$	33.5	24.3	15.1	22.8	23.9$_{\pm 0.3}$
Dr.GRPO (Liu et al., 2025)	72.8	54.3	46.6	33.8	24.1	15.3	22.6	23.9
VAPO (Yue et al., 2025)	73.5	54.8	47.0	34.2	24.8	15.7	23.1	24.5
DAPO (Yu et al., 2025)	74.2$_{\pm 0.4}$	55.3$_{\pm 0.7}$	47.6$_{\pm 0.8}$	35.2	25.7	16.3	23.9	25.3$_{\pm 0.3}$
GSPO (Zheng et al., 2025)	73.8$_{\pm 0.5}$	54.9$_{\pm 0.8}$	48.5$_{\pm 0.9}$	34.8	25.4	16.1	25.8	25.5$_{\pm 0.3}$
Memory- and skill-augmented								
EvolveR + GRPO (Wu et al., 2025)	73.6	54.8	46.3	34.5	25.1	15.4	23.5	24.6
SimpleMem + GRPO (Liu et al., 2026)	73.1	54.5	46.9	33.9	24.6	16.5	23.1	24.5
ARISE + GRPO	75.6$_{\pm 0.4}$$_{\uparrow 2.7}$	56.9$_{\pm 0.6}$$_{\uparrow 2.8}$	48.3$_{\pm 0.7}$$_{\uparrow 1.8}$	37.4$_{\uparrow 3.9}$	27.0$_{\uparrow 2.7}$	17.9$_{\uparrow 2.8}$	25.7$_{\uparrow 2.9}$	27.0$_{\pm 0.3}$$_{\uparrow 3.1}$
ARISE + DAPO	**76.9**$_{\pm 0.5}$$_{\uparrow 4.0}$	**57.5**$_{\pm 0.7}$$_{\uparrow 3.4}$	**50.4**$_{\pm 0.8}$$_{\uparrow 3.9}$	**38.6**$_{\uparrow 5.1}$	**28.1**$_{\uparrow 3.8}$	**18.8**$_{\uparrow 3.7}$	**27.5**$_{\uparrow 4.7}$	**28.3**$_{\pm 0.3}$$_{\uparrow 4.4}$
Phi-4-mini-instruct-3.8B								
Base model	36.3	9.7	4.3	10.3	7.1	4.4	6.5	7.1
GRPO family								
GRPO (Guo et al., 2025)	42.7	10.3	4.7	13.2	9.5	6.0	8.8	9.4
Dr.GRPO (Liu et al., 2025)	42.5	10.6	4.6	13.4	9.3	6.1	8.6	9.4
VAPO (Yue et al., 2025)	43.1	10.8	4.9	13.8	9.8	6.3	9.1	9.8
DAPO (Yu et al., 2025)	44.1	11.5	6.1	14.6	10.7	7.0	9.5	10.5
GSPO (Zheng et al., 2025)	43.6	11.8	5.0	14.2	10.3	6.7	9.8	10.2
Memory- and skill-augmented								
EvolveR + GRPO (Wu et al., 2025)	43.2	10.9	5.0	14.0	10.1	6.2	9.4	9.9
SimpleMem + GRPO (Liu et al., 2026)	43.0	10.7	4.9	13.6	9.8	6.4	9.0	9.7
ARISE + GRPO	45.4$_{\uparrow 2.7}$	12.6$_{\uparrow 2.3}$	6.0$_{\uparrow 1.3}$	16.1$_{\uparrow 2.9}$	11.4$_{\uparrow 1.9}$	8.0$_{\uparrow 2.0}$	10.7$_{\uparrow 1.9}$	11.6$_{\uparrow 2.2}$
ARISE + DAPO	**46.8**$_{\uparrow 4.1}$	**13.4**$_{\uparrow 3.1}$	**7.2**$_{\uparrow 2.5}$	**16.9**$_{\uparrow 3.7}$	**12.5**$_{\uparrow 3.0}$	**8.6**$_{\uparrow 2.6}$	**11.5**$_{\uparrow 2.7}$	**12.4**$_{\uparrow 3.0}$

† Macro average over the four Omni-MATH domains. Omni-MATH sub-categories follow the official domain taxonomy.

4.2 Performance

ARISE+DAPO attains the highest value in each column of Table 3. ARISE+GRPO surpasses the non-ARISE methods on six of the eight Qwen3-4B columns and seven of the eight Phi-4-mini columns despite a weaker objective, with GSPO and DAPO keeping edges of at most 0.2 that ARISE+DAPO reverses. The skill mechanism therefore composes with the host objective across two backbones of different scale and family, and the skill cycle substitutes in part for the refinements that separate DAPO and GSPO from vanilla GRPO. Across three training seeds, the ARISE gains over their host optimizers exceed the seed standard deviation. On the Omni-MATH average, the gains of 3.1 points for ARISE+GRPO and 3.0 for ARISE+DAPO are an order of magnitude above the deviation of 0.3, with a paired t-test over seeds giving $p < 0.01$. On the four seeded Qwen3-4B columns, DAPO improves over GRPO by 1.1 to 1.4 points, whereas composing ARISE with GRPO yields 1.8 to 3.1.

Gains are uniformly larger on Omni-MATH and peak in Algebra, followed by Geometry on Qwen3-4B, where reusable patterns such as common-base rewriting, telescoping, and auxiliary-line construction transfer across instances. Number Theory and Combinatorics, whose solutions hinge on problem-specific insight, show smaller but consistent improvements. The memory-augmented baselines improve over GRPO but fail to reach the DAPO or GSPO level, and the adapted SAGE of Appendix G.1 lands in the same range, since retrieval is decoupled from policy optimization. By routing selection through the parameters that generate solutions, ARISE ties the advantage signal to the selection decision, so skill augmentation compounds with the underlying refinement, and ARISE+DAPO leads ARISE+GRPO in each column of both backbones.

4.3 Ablation and Analysis

Table 4 isolates each design choice on Qwen3-4B, where collapsing the trajectory reward to binary (a), so that a correct solution receives ζ_1 whether or not a skill was selected, causes the largest drop and skill utilization falls from 73% to 28%. Random selection (b) keeps utilization high yet degrades

Table 4: The table reports ablations on Qwen3-4B in Pass@1 (%).

	Variant	AIME 24	AIME 25	Omni
	ARISE (full)	**56.9**	**48.3**	**27.0**
(a)	Binary R^w	55.0	46.8	25.1
(b)	Random select.	55.4	47.2	25.8
(c)	w/o τ_{G+1}	55.2	46.9	25.4
(d)	FIFO eviction	56.3	47.8	26.4
(e)	Self-rollout concl.	55.8	47.5	26.1
(f)	Decoupled scoring head	55.3	47.1	25.7

Table 5: The table tracks the R^s distribution (%) over training.

Stage	r_1	r_2	r_3	r_4
Phase I (early)	8.2	51.3	14.6	25.9
Phase I (late)	5.1	38.7	24.8	31.4
Phase II (early)	3.8	29.4	31.2	35.6
Phase II (late)	2.3	22.1	34.8	40.8

accuracy, so the scoring of Eq. (4) captures query-skill matching, not a generic benefit of arbitrary augmentation. Removing the verification rollout (c) removes both the quality level and the initial utility of new entries and gives the lowest cumulative skill production of the variants. First-in-first-out eviction (d) affects accuracy least but drives utilization to 88%, since the policy then selects with quality-based filtering disabled. Concluding from the policy's own correct rollouts instead of the ground-truth trace (e) produces broader but less targeted skills, yet still improves over GRPO by 1.7 points on AIME 2024 and 2.2 on Omni-MATH. Ablation (f) replaces the scorer with a decoupled head with its own parameters ϕ, trained by REINFORCE on the same trajectory reward, which keeps the option mechanism but severs the shared-parameter coupling. The head falls 1.2 to 1.6 points below the full system beyond the seed deviation (Appendix G.2) and lands within 0.1 point of random selection (b), as Corollary B.1 predicts for a head whose gradient vanishes under group normalization.

The per-step cost in Appendix F.3 is 64.9s for ARISE against 51.2s for GRPO on 2×H200, an overhead of 26.8% located in skill conclusion and the verification rollout, and it does not grow with the library, since only cache entries enter the scoring pass. In Table 5 the neutral r_2 outcome dominates early in Phase I, the combined $r_3 + r_4$ fraction rises from 40.5% to 75.6% over training, and harmful r_1 skills decline steadily. Appendix H.2 finds the cache share of entries concluded early in Phase I falling from 100% to 11% while accuracy rises, so weak early skills are displaced.

Figure 2 examines the cost and dynamics of the skill mechanism. Panel (a) plots accuracy gain against token overhead, and EvolveR pays a large overhead for a marginal gain through lengthy experience narratives, whereas ARISE+GRPO and ARISE+DAPO land in the high-gain region with efficiency ratios about eight times higher, since each selected skill is a short document gated by the confidence of μ_θ. Panel (b) shows ARISE pulling ahead of GRPO as concluded skills accumulate, with the gap widening through step 1,500 and accuracy continuing to climb after the cumulative count plateaus. Panel (c) fixes the budget at 43 hours on 2×H200, the cost of 3,000 GRPO steps, of which ARISE uses 36 hours for its 2,000 steps: baselines at $G = 8$ complete more steps and their $G = 12$ counterparts fewer (Liu et al., 2025), yet ARISE+DAPO surpasses $G = 12$ DAPO with a smaller FLOP count, so the gains come from the skill mechanism.

Generalization across scale and domain. The gain persists on a third backbone, Qwen3-8B-Base, where ARISE+GRPO improves over GRPO from 78.4 to 80.6 on AMC 2023, from 33.6 to 35.9 on AIME 2024, and from 30.1 to 32.8 on Omni-MATH under the identical protocol (Appendix G.3). On code generation with unit-test pass as the reward, Qwen3-4B-Instruct-2507 trains on the DeepCoder corpus (Luo et al., 2025a) with the official solution as the ground-truth trace, the conclusion template adapted to algorithmic-pattern fields, and the remaining settings untouched (Appendix I). Over three seeds, ARISE+GRPO improves Pass@1 from 78.7 to 80.5 on HumanEval (Chen et al., 2021) and from 73.2 to 75.6 on MBPP (Austin et al., 2021), several times the seed deviation.

5 Conclusion

We presented ARISE, a hierarchical reinforcement learning framework that casts each library entry as an option and unifies skill selection, generation, and policy optimization under shared parameters, so that the skill library co-evolves with the agent. Across two base models, seven benchmarks, a third backbone, and code generation, ARISE improves over GRPO-family and skill-augmented baselines, and ablations identify the third reward level and the shared-parameter coupling as the drivers of skill adoption. Extending ARISE to multi-turn tool-use agents, where an option may span several steps, is future work, and Appendix J records the limitations.

AI USE STATEMENT

In this work, we used generative AI tools only to polish the writing of the manuscript. We have not used generative AI tools for developing the theoretical model or conceptual framework, formulating mathematical claims, designing the methodology or experiments, implementing the method, or interpreting the results, and the remaining tasks with required disclosure are not applicable to this work. We have reviewed all AI-assisted text and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies mathematical reasoning and code generation on publicly available datasets and benchmarks. It is free of human subjects, personal data, and sensitive content, and we find it compliant with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The training corpus, the base models, the baselines, the benchmarks, and the evaluation protocol are specified in §4.1, and the hyperparameters of ARISE together with the shared GRPO settings are listed in Table 7 of Appendix C.1. Appendix C.2 gives the per-step training pipeline, Appendix C.3 the decoding configuration, the 32-sample averaging, and the three-seed statistics, Appendix D the skill document schema, the injection format, and the cache-reservoir maintenance rules, and Appendix E the skill conclusion prompt, the validation pipeline, and the numerical reward levels. The adapted SAGE baseline, the decoupled scoring head, the third backbone, and the code-generation setup are described in Appendices G.1, G.2, G.3, and I. The source code, including the training scripts and the evaluation harness, is provided in the repository linked in the abstract.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Md Tanvirul Alam and Nidhi Rastogi. Limits of generalization in rlvr: Two case studies in mathematical reasoning. *arXiv preprint arXiv:2510.27044*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 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*, 2025.

Ce Hao, Catherine Weaver, Chen Tang, Kenta Kawamoto, Masayoshi Tomizuka, and Wei Zhan. Skill-critic: Refining learned skills for hierarchical reinforcement learning. *IEEE Robotics and Automation Letters*, 9(4):3625–3632, 2024.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Yu Li, Tian Lan, and Zhengling Qi. When right meets wrong: Bilateral context conditioning with reward-confidence correction for grpo. *arXiv preprint arXiv:2603.13134*, 2026.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level, 2025a. Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025b. Notion Blog.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Dang Nguyen, Viet Dac Lai, Seunghyun Yoon, Ryan A Rossi, Handong Zhao, Ruiyi Zhang, Puneet Mathur, Nedim Lipka, Yu Wang, Trung Bui, et al. Dynasaur: Large language agents beyond predefined actions. *arXiv preprint arXiv:2411.01747*, 2024.

Avinash Patil and Aryan Jadon. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*, pp. 212–223. Springer, 2002.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43, 2025.

Richard Sutton. Intra-option learning about temporally abstract actions. In *Proceedings of the Fifteenth International...*, 1998.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, pp. 3540–3549. PMLR, 2017.

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.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. *arXiv preprint arXiv:2512.17102*, 2025.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, et al. Agentic reasoning for large language models. *arXiv preprint arXiv:2601.12538*, 2026.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

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.

Fengli Xu, Qianyue Hao, Chenyang Shao, Zefang Zong, Yu Li, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, et al. Toward large reasoning models: A survey of reinforced reasoning with large language models. *Patterns*, 6(10), 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

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.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Chenyu Zhou, Huacan Chai, Wenteng Chen, Zihan Guo, Rong Shan, Yuanyi Song, Tianyi Xu, Yingxuan Yang, Aofan Yu, Weiming Zhang, et al. Externalization in llm agents: A unified review of memory, skills, protocols and harness engineering. *arXiv preprint arXiv:2604.08224*, 2026.

APPENDIX

A NOTATION

Table 6 collects the symbols used throughout §3 and the appendix.

Table 6: The table collects the running notation of ARISE. The trajectory reward R^w is the only reward that enters the policy gradient, and the skill quality level R^s governs library maintenance alone.

Symbol	Meaning	Symbol	Meaning
q_t, τ_t^+	query and its ground-truth trace	z_t	option selected for q_t, shared by the G rollouts
$L_t = C_t \cup R_t$	library, active cache, reservoir	$\tilde{m}_t, \tau_{G+1}$	concluded skill and its verification rollout
$s_t = (q_t, L_t)$	augmented state	$R_i^w \in \{\zeta_0, \zeta_1, \zeta_2\}$	trajectory reward entering the advantage
$m_k, o_k, o_\varnothing$	skill document, induced option, unaided fallback	$R^s \in \{r_1, \dots, r_4\}$	skill quality level initializing utility
$\mu_\theta(o_k \mid s_t)$	policy over options	$\bar{r}_k, n_k$	running utility and selection count of entry k
$\pi_{\theta,k}(\tau \mid q)$	intra-option policy conditioned on m_k	$\hat{A}_i$	group-relative advantage of rollout i

B THEORETICAL ANALYSIS

B.1 PROOFS OF PROPOSITIONS 3.1 AND 3.2

Proof of Proposition 3.1. The episode structure of §3.1 contains one decision epoch: the state $s_t = (q_t, L_t)$ is drawn, μ_θ selects z_t, the intra-option policy generates τ_t, and the episode terminates when β_{z_t} fires at the end-of-sequence token. The objective therefore decomposes over the option set,

$$J(\theta) \;=\; \mathrm{E}_{s_t}\Big[\sum_k \mu_\theta(o_k \mid s_t)\, V_k(s_t)\Big], \qquad V_k(s_t) \;=\; \mathrm{E}_{\tau\sim\pi_{\theta,k}(\cdot|q_t)}\big[R^w(\tau)\big]. \tag{10}$$

The option-critic gradient (Bacon et al., 2017) comprises three terms, namely the gradient through the policy over options weighted by option values, the intra-option term of the intra-option policy gradient theorem, and the termination term proportional to $\nabla_\theta \beta$. The termination β_k is the indicator that the last generated token equals the end-of-sequence symbol, a fixed function of the generated trajectory that does not depend on θ, so $\nabla_\theta \beta_k = 0$ pointwise and the termination term vanishes identically. Because each option spans one full trajectory and the episode ends at its termination, the semi-MDP Bellman recursion truncates after the single epoch, the option value equals the per-query return, $Q_\Omega(s_t, o_k) = V_k(s_t)$, and the continuation value beyond termination equals zero. Differentiating Eq. (10) with the product rule gives

$$\nabla_\theta J(\theta) = \mathrm{E}_{s_t}\Big[\sum_k V_k(s_t)\, \nabla_\theta \mu_\theta(o_k \mid s_t)\Big] + \mathrm{E}_{s_t}\Big[\sum_k \mu_\theta(o_k \mid s_t)\, \nabla_\theta V_k(s_t)\Big], \tag{11}$$

and the log-derivative identities $\nabla_\theta \mu_\theta = \mu_\theta \nabla_\theta \log \mu_\theta$ and

$$\nabla_\theta V_k(s_t) = \sum_\tau R^w(\tau)\, \nabla_\theta \pi_{\theta,k}(\tau \mid q_t) = \mathrm{E}_{\tau\sim\pi_{\theta,k}}\big[R^w(\tau)\, \nabla_\theta \log \pi_{\theta,k}(\tau \mid q_t)\big] \tag{12}$$

turn the two sums into one expectation over the sampled option and trajectory,

$$\nabla_\theta J(\theta) = \mathrm{E}_{s_t,\, z_t\sim\mu_\theta,\, \tau_t\sim\pi_{\theta,z_t}}\Big[R^w(\tau_t)\,\big(\nabla_\theta \log \mu_\theta(z_t \mid s_t) + \nabla_\theta \log \pi_{\theta,z_t}(\tau_t \mid q_t)\big)\Big]. \tag{13}$$

For a per-state baseline $b(s_t)$, the identities $\sum_k \nabla_\theta \mu_\theta(o_k \mid s_t) = \nabla_\theta 1 = 0$ and $\mathrm{E}_\tau[\nabla_\theta \log \pi_{\theta,k}(\tau \mid q_t)] = 0$ make the substitution $R^w \mapsto R^w - b(s_t)$ exact, and the group statistics of Eq. (7) realize the baseline together with the division by $s_G + \epsilon$, the standard critic-free rescaling of GRPO, which yields Eq. (5). The confidence gate substitutes $o_\varnothing$ on the event $\max_k \mu_\theta(o_k \mid s_t) < \delta$, whose indicator is piecewise constant in θ, so the gate leaves the gradient unchanged outside a boundary set of measure zero. □

Proof of Proposition 3.2. The group Monte Carlo estimate of the selection term factorizes because the selected option is shared by the G rollouts,

$$\frac{1}{G}\sum_{i=1}^{G} \hat{A}_i\, \nabla_\theta \log \mu_\theta(z_t \mid s_t) \;=\; \Big(\frac{1}{G}\sum_{i=1}^{G} \hat{A}_i\Big) \nabla_\theta \log \mu_\theta(z_t \mid s_t), \quad \sum_{i=1}^{G} \hat{A}_i = \frac{\sum_i R_i^w - G\bar{R}_t^w}{s_G + \epsilon} = 0, \tag{14}$$

so the estimate equals zero for every realized group and the selection signal travels through the shared parameters. At $\theta = \theta'$ every importance ratio $\rho_{i,l}$ equals one, the two branches of the clipped surrogate of Eq. (8) coincide, and the per-token derivative equals $\hat{A}_i \nabla_\theta \log \pi_\theta(x_{i,l} \mid c_{i,<l})$, so

$$\nabla_\theta \mathcal{L}_G(\theta)\big|_{\theta=\theta'} = \frac{1}{G}\sum_{i=1}^{G} \frac{\hat{A}_i}{|\tau_i|} \sum_{l=1}^{|\tau_i|} \nabla_\theta \log \pi_\theta(x_{i,l} \mid c_{i,<l}) = \frac{1}{G}\sum_{i=1}^{G} \hat{A}_i \, g_i. \tag{15}$$

The ascent step $\theta^+ = \theta + \eta \, \nabla_\theta \mathcal{L}_G$ satisfies $\|\theta^+ - \theta\| = O(\eta)$, and expanding the score to first order around θ gives

$$\ell_{\theta^+}(o_k \mid s_t) = \ell_\theta(o_k \mid s_t) + \big\langle h_k, \, \theta^+ - \theta \big\rangle + O\big(\|\theta^+ - \theta\|^2\big), \tag{16}$$

and substituting Eq. (15) yields Eq. (6). Propagating the score changes $\Delta\ell_k$ of Eq. (6) through the softmax of Eq. (4) moves the selection probabilities by

$$\mu_{\theta^+}(o_k \mid s_t) - \mu_\theta(o_k \mid s_t) = \frac{\mu_\theta(o_k \mid s_t)}{\sigma}\Big(\Delta\ell_k - \sum_{j=1}^{|C_t|} \mu_\theta(o_j \mid s_t)\, \Delta\ell_j\Big) + O(\eta^2), \tag{17}$$

so probability mass moves toward the entries whose kernel alignment with the high-advantage rollouts exceeds the cache average. □

Corollary B.1. *With selection scored by a head whose parameters ϕ are disjoint from θ, the score change through the policy update equals zero at every order. Under the group-relative advantage the REINFORCE estimate of the head also vanishes for every realized group, so the head receives zero gradient over the whole batch and its selection distribution stays at initialization.*

Proof. The head's scores do not depend on θ, so every term of the expansion above equals zero. The factorization of the selection estimate applies verbatim to $\nabla_\phi \log \mu_\phi(z_t \mid s_t)$ because z_t is shared by the group, and the batch gradient sums per-group estimates that each equal zero. □

The corollary predicts that the decoupled head of Appendix G.2 selects independently of outcomes, and Table 4 bears the prediction out, since the head lands within 0.1 point of random selection on the three columns. Such a head can learn from outcomes only through a signal that survives group normalization, such as the group-mean reward against a batch-level baseline, whereas the shared-parameter channel of Eq. (6) reaches selection through the within-group advantages themselves.

B.2 Posterior Reading of the Selection Rule

Under a uniform prior $p(o_k) = 1/|C_t|$ over cache entries, the posterior over which document the policy would articulate for q_t is

$$p(o_k \mid q_t) = \frac{\pi_\theta(m_k \mid q_t)}{\sum_j \pi_\theta(m_j \mid q_t)} = \frac{\exp\big(|m_k|\, \ell_\theta(o_k \mid s_t)\big)}{\sum_j \exp\big(|m_j|\, \ell_\theta(o_j \mid s_t)\big)}, \tag{18}$$

where the second equality follows from $\log \pi_\theta(m_k \mid q_t) = |m_k|\, \ell_\theta(o_k \mid s_t)$ in the notation of Eq. (4). Additional tokens contribute negative log-probabilities, so at equal per-token fit the exact posterior prefers the shortest document. Eq. (4) replaces the entry-specific exponent $|m_k|$ by the common exponent $1/\sigma$, which tempers each entry of Eq. (18) with exponent $1/(\sigma|m_k|)$, removes the length preference, and preserves the ranking induced by per-token fit. For a cache of equal-length documents, the two rules coincide up to the temperature $\sigma|m|$. The selection rule therefore performs tempered posterior inference over which cached reasoning pattern the policy is most inclined to reproduce for the query, with small σ approaching the greedy choice and large σ approaching the prior.

B.3 Fixed Point of the Running Utility

The running utility of §3.2 interpolates between the source-query level and the rewards of later selections, and Proposition B.1 quantifies the interpolation.

Proposition B.1 (Utility fixed point)**.** *Fix the policy and the selection distribution between library changes, and let the group-mean trajectory rewards $\bar{R}^{w,(1)}, \ldots, \bar{R}^{w,(n)}$ observed at the successive selections of entry k be independent with mean ν_k and variance ς^2. After n selections the running utility satisfies*

$$\bar{r}_k^{(n)} = \lambda^n R^s + (1-\lambda) \sum_{j=1}^{n} \lambda^{n-j}\, \bar{R}^{w,(j)}, \qquad \mathrm{E}\big[\bar{r}_k^{(n)}\big] = \lambda^n R^s + \big(1-\lambda^n\big)\nu_k, \tag{19}$$

with variance $\varsigma^2(1-\lambda)\big(1-\lambda^{2n}\big)/(1+\lambda)$.

Proof. The UPDATE rule gives $\bar{r}_k^{(1)} = \lambda R^s + (1-\lambda)\bar{R}^{w,(1)}$, and if the unrolled form of Eq. (19) is valid after $n-1$ selections, one further application of the rule gives

$$\bar{r}_k^{(n)} = \lambda\Big(\lambda^{n-1} R^s + (1-\lambda) \sum_{j=1}^{n-1} \lambda^{n-1-j}\, \bar{R}^{w,(j)}\Big) + (1-\lambda)\bar{R}^{w,(n)}, \tag{20}$$

which reproduces Eq. (19). The expectation follows by linearity with $(1-\lambda)\sum_{j=1}^{n} \lambda^{n-j} = 1-\lambda^n$, and independence gives the variance $(1-\lambda)^2\varsigma^2 \sum_{j=1}^{n} \lambda^{2(n-j)} = \varsigma^2(1-\lambda)^2\big(1-\lambda^{2n}\big)/\big(1-\lambda^2\big)$, which simplifies through $1-\lambda^2 = (1-\lambda)(1+\lambda)$. □

The initialization enters with weight λ^n, 0.35 after ten selections at $\lambda = 0.9$, and the utility converges to ν_k, the expected group-mean trajectory reward conditioned on selecting entry k, with asymptotic variance one nineteenth of the per-group variance. EVICT, LOAD, and DELETE therefore order entries by estimates of conditional value on later queries, and the source-query level governs an entry only until selections accumulate, consistent with the displacement measurements of Appendix H.2. During training ν_k drifts with the policy, and the exponential window tracks the drift with an effective horizon of $1/(1-\lambda) = 10$ selections.

C Implementation Details

C.1 Training Configuration and Hyperparameters

The methods are trained on the DeepScaleR dataset (Luo et al., 2025b), which supplies approximately 40K problem-solution pairs drawn from AMC, AIME, MATH, and OlympiadBench, for 2,000 optimization steps on 2×H200 GPUs. The reference solution attached to each problem serves as the ground-truth trace τ_t^+ from which skill conclusion draws in Eq. (2). The methods use group size $G = 8$, clipping width $\epsilon_c = 0.2$, and an effective batch of 64 queries per policy update. The optimizer is AdamW with learning rate 1×10^{-6}, weight decay 0.01, and a cosine schedule preceded by 50 linear warm-up steps. Solution rollouts $\tau_1, \ldots, \tau_G$ and the verification rollout τ_{G+1} share a maximum length of 4,096 tokens, while the conclusion rollout that produces $\tilde{m}_t$ is capped at 192 new tokens. Dynamic sampling, which removes groups in which the G trajectories receive the same reward before advantage estimation (Yu et al., 2025), is part of the DAPO objective and therefore applies only to the DAPO row and the ARISE+DAPO row. ARISE+GRPO and the GRPO row use the plain objective and keep the full batch of groups. Phase I spans the first $N_w = 500$ steps, and Phase II activates the policy over options at step 501. Table 7 lists the hyperparameters introduced by ARISE together with the settings shared with the baselines.

C.2 Per-Step Pipeline

Each training step executes the sequence below, where steps 2 and 3 execute only in Phase II and the remaining steps execute in both phases.

1. **Batch loading.** Sample a mini-batch of pairs $(q_t, \tau_t^+) \sim \mathcal{D}$.
2. **Option selection.** The policy over options scores the entries of the active cache C_t through Eq. (4) and samples z_t from the resulting distribution, falling back to $o_\varnothing$ when $\max_k \mu_\theta(o_k \mid s_t) < \delta$.

Table 7: The table lists the hyperparameters of ARISE. Rows above the double rule are introduced by the method, and rows below it are shared with the baselines.

Hyperparameter	Symbol	Value
Policy over options		
Selection temperature	σ	1.0
Confidence threshold	δ	0.35
Document tokens used for scoring		128
Skill library		
Cache capacity	κ_c	10
Reservoir capacity	κ_r	100
Utility EMA decay	λ	0.9
Skill conclusion		
Sampling temperature		0.7
Top-p		0.95
Maximum new tokens		192
Maximum document length		220 characters
Trajectory reward		
Reward levels	$\zeta_0, \zeta_1, \zeta_2$	0, 1, 2
Skill quality reward		
Reward levels	r_1, r_2, r_3, r_4	-0.5, 0, 0.5, 1
Training schedule		
Warm-up steps	N_w	500
Shared GRPO settings		
Group size	G	8
Rollout sampling temperature		0.7
Rollout top-p		0.95
Clipping width	ϵ_c	0.2
Learning rate		1×10^{-6}
Weight decay		0.01
Maximum generation length		4,096 tokens

3. **Prompt construction.** The selected document m_{z_t} is prepended to q_t in the format of Appendix D.2, and under the fallback branch the query is presented unmodified.
4. **Solution rollouts.** Draw G trajectories from $\pi_{\theta', z_t}(\cdot \mid q_t)$ under the parameters θ' frozen before the current update.
5. **Trajectory reward.** Grade each trajectory against the ground-truth answer $a^\star(q_t)$ and assign $R_i^w \in \{\zeta_0, \zeta_1, \zeta_2\}$ through Eq. (9), where the level ζ_2 is reachable only under a selected skill.
6. **Dynamic sampling (DAPO rows only).** Remove groups in which the G trajectories receive the same reward, following DAPO (Yu et al., 2025).
7. **Advantage estimation and update.** Compute $\hat{A}_i$ through Eq. (7) and update θ through Eq. (8), which realizes both terms of Eq. (5) under the shared parameters.
8. **Skill conclusion.** Draw $\tilde{m}_t \sim \pi_\theta(\cdot \mid q_t, \tau_t^+)$ as an inference-only step and validate the output as described in Appendix E.1.
9. **Verification rollout.** Solve q_t once more conditioned on $\tilde{m}_t$, compare the outcome with the G solution rollouts, and assign R^s as described in Appendix E.2.
10. **Library maintenance.** Refresh the utility of the selected entry with the group mean $\bar{R}_t^w$, then execute ADD, EVICT, LOAD, and DELETE in order as described in Appendix D.3.

C.3 EVALUATION PROTOCOL

The results report Pass@1 averaged over 32 evaluation samples drawn under fixed weights, each generating one solution per problem with sampling temperature 0.7 and top-p 0.95, the rollout configuration of Table 7, so the 32-sample average captures decoding variance, not training variance.

The full ARISE pipeline stays active at evaluation time: the policy over options scores the final cache C through Eq. (4) and samples an option under the training temperature σ. The intra-option policy generates the solution conditioned on the selected document, and the query is solved unaided whenever the confidence gate rejects the candidates. Answers on AMC 2023, AIME 2024, and AIME 2025 are verified by exact numerical match, and Omni-MATH follows the official protocol of Gao et al. (2024) with symbolic equivalence checking. Baselines and ablation variants share the decoding configuration and the 32-sample averaging.

Training variance is measured by retraining: the Qwen3-4B configurations GRPO, DAPO, GSPO, adapted SAGE, the decoupled scoring head, ARISE+GRPO, and ARISE+DAPO were retrained with three independent training seeds, and Table 3, Table 13, and Table 14 report the mean and the standard deviation across seeds. Paired t-tests over the three seeds assess each ARISE gain against its host optimizer. On the Omni-MATH average, the tests give $p < 0.01$ for both ARISE+GRPO over GRPO and ARISE+DAPO over DAPO, and the ARISE gains in Table 3 exceed their seed standard deviations.

D Skill Library

D.1 Document Schema

Library entries are stored as JSON documents with the five fields named in §3.2, so that the score ℓ_θ of Eq. (4) reflects content, not format variation. Table 8 lists the constraint on each field, and Figure 3 shows a concluded document.

Table 8: The table gives the skill document schema with the constraint on each field.

Field	Maximum length	Content
`skill_name`	40 chars	Snake_case identifier, required to be non-empty
`problem_type`	one label	One of algebra, geometry, combinatorics, number_theory, calculus, general
`key_insight`	160 chars	Core mathematical idea stating why the approach works
`method`	3 steps	Two or three procedural steps of at most 100 characters each
`check`	100 chars	Verification procedure, defaulting to "Substitute back to verify"
Whole document		Hard-capped at 220 characters

```
{
"skill_name": "exponential_base_matching",
"problem_type": "algebra",
"key_insight": "Write both sides as powers of one
base, then equate exponents",
"method": [
"Rewrite both sides in a common base",
"Equate exponents and solve",
"Check the root in the original"
],
"check": "Substitute back to verify"
}
```

Figure 3: The skill document shown here follows the schema of Table 8 and was concluded from an exponential-equation problem.

D.2 Option Scoring and Injection

For each entry $m_k \in C_t$, the policy over options feeds the concatenation of q_t and m_k through π_θ and averages the per-token log-probabilities of m_k to obtain $\ell_\theta(o_k \mid s_t)$. Documents are truncated to 128 tokens for scoring, while the full document is used for injection. The scores of the $\kappa_c = 10$ candidates come from one batched forward pass over the query-document concatenations. Since a document occupies roughly 60 to 80 tokens after tokenization and a query 50 to 150 tokens, the pass processes at most $10 \times 278 = 2{,}780$ tokens, which accounts for the 1.8s selection cost in Table 11.

The selected document is injected by prepending it to the query with a `SKILL:` prefix inside the chat template:

```
<|im_start|>user
SKILL:{...skill JSON...}
<math question>
<|im_end|>
<|im_start|>assistant
```

Under the fallback branch $o_\varnothing$ the query is presented unprefixed, and the resulting trajectories are graded under the same solution reward R^w.

D.3 Cache and Reservoir Maintenance

The library $L_t = C_t \cup R_t$ consists of an active cache C_t with capacity $\kappa_c = 10$, which induces the option set $\Omega(C_t)$ scored once per training step, and a reservoir R_t with capacity $\kappa_r = 100$ that archives displaced entries for later promotion. Each entry carries its document m_k, its running reward $\bar{r}_k$, and its cumulative selection count n_k. The five operations of §3.2 execute in the fixed order Update, Add, Evict, Load, Delete at the end of each step.

- Update applies $\bar{r}_{z_t} \leftarrow \lambda\, \bar{r}_{z_t} + (1-\lambda)\, \bar{R}^w_t$ to the entry selected in the current step with $\lambda = 0.9$, where $\bar{R}^w_t$ is the mean trajectory reward of the group that invoked the entry, and entries that were not selected keep their previous value.
- Add inserts $\tilde{m}_t$ into C_t with $\bar{r}_{\tilde{m}_t}$ initialized to the level R^s of its verification rollout and $n_{\tilde{m}_t} = 0$, and is skipped when validation rejects the concluded document.
- Evict moves the lowest-utility cache entry to R_t whenever the cache exceeds κ_c after insertion, preserving $\bar{r}_k$ and n_k.
- Load swaps the highest-utility reservoir entry with the lowest-utility cache entry whenever the reservoir entry carries the higher utility, so that entries concluded early can re-enter the active pool once the policy can exploit them.
- Delete removes reservoir entries with $\bar{r}_k < u_{10}(R_t)$ and $n_k = 0$, where $u_{10}(R_t)$ is the 10th percentile of reservoir utilities.

The order places Update before Add so that the selected entry reflects the current step before insertion can trigger an eviction, and places Load after Evict so that a just-evicted entry does not re-enter in the same step. When the confidence gate rejects the candidates, Update is skipped and the remaining operations proceed on the unchanged cache. When the reservoir is empty, Load and Delete leave the library unchanged. When dynamic sampling removes a zero-variance group, skill conclusion, verification, and maintenance still proceed for that query.

E Skill Conclusion and Marginal Reward

E.1 Conclusion Prompt and Validation

After the G solution rollouts, the policy concludes $\tilde{m}_t \sim \pi_\theta(\cdot \mid q_t, \tau_t^+)$ through one inference-only rollout conditioned on the query and its ground-truth trace, using the prompt of Figure 4 with sampling temperature 0.7, top-p 0.95, and at most 192 new tokens.

Conclusion Prompt Template

```
You are a skill distiller for math reasoning.
Given one question and its reference solution,
summarize ONE reusable skill.

Question: {question}
Reference solution: {ground_truth_trace}

Output MUST be a valid JSON object and nothing else
(no markdown/code fences). Schema:
{"skill_name", "problem_type", "key_insight",
"method", "check"}

Rules:
- Be generic and transferable; do not copy specific numbers.
- Keep the whole skill within 220 characters.
- key_insight is the most important field.
- method must contain 2-3 concise steps.
- Focus on improving correctness, not style.
```

Figure 4: The prompt shown here conditions the conclusion rollout on the query and its ground-truth trace, with the trace clipped to 800 characters.

The generated text passes through four stages before entering the library. Extraction isolates the JSON object from the raw generation and strips surrounding text or fences, and parsing verifies that the five fields exist and satisfy the constraints of Table 8. Truncation clips fields that exceed their character limit and discards the document when the total still exceeds 220 characters, and insertion hands the validated document to ADD. When parsing fails, a fallback constructs a minimal document whose key insight consists of the opening of the ground-truth trace, so that the query still contributes an entry. The primary path succeeds in over 85% of attempts after warm-up, and the fallback rate declines through Phase II as π_θ improves at structured generation. The 220-character cap also bounds the prompt overhead of injection at roughly 66 tokens per query, as measured in Appendix F.3.

E.2 REWARD LEVELS AND RUNNING UTILITY

The verification rollout τ_{G+1} solves q_t conditioned on $\tilde{m}_t$ under the same decoding configuration as the solution rollouts, and its correctness $v(\tau_{G+1}, q_t)$ is combined with the G solution outcomes to assign one of the four levels of §3.3. The numerical values $r_1 = -0.5$, $r_2 = 0$, $r_3 = 0.5$, and $r_4 = 1$ place the neutral outcome at zero, so a harmful skill enters the cache with negative utility and a critical one with the same value as a correct unaided solution under R^w. Under the Phase I level compositions of Table 5, the mean initial utility of a concluded entry lies between 0.29 and 0.41, consistent with the 0.31 recorded at the phase boundary in Table 17. The level does not enter the policy gradient: it initializes $\bar{r}_{\tilde{m}_t}$ when the entry enters the cache. UPDATE then refreshes the utility with the mean trajectory reward of the later groups that select the entry, and the utility so maintained orders entries for EVICT, LOAD, and DELETE. The source-query estimate is therefore discounted by $\lambda = 0.9$ at each selection, and within a few selections the utility reflects performance on the later queries the entry is applied to. For an entry with zero selections, the source-query level remains the value that governs EVICT, and the effect of an inflated initial value stays bounded. The entry competes against the subsequently concluded skills, of which two to three enter per hundred steps in Phase II, so an entry whose value does not translate into selection is displaced within a few hundred steps. The cache capacity $\kappa_c = 10$ limits how many such entries can coexist, and the confidence gate δ keeps an entry with zero selections out of the rollouts. Table 5 in the main text tracks how the four levels redistribute over training, and Appendix H.1 recomputes the level on unseen queries.

F Extended Analysis

F.1 Hyperparameter Sensitivity

Table 9 varies the confidence threshold δ, the cache capacity κ_c, and the warm-up length N_w one at a time on Qwen3-4B-Instruct-2507, reporting Pass@1 on AIME 2024 with the remaining settings at their defaults.

Table 9: Each block of the table varies one hyperparameter around its underlined default and reports Pass@1 (%) on AIME 2024 for Qwen3-4B.

Confidence threshold δ	0.15	0.25	<u>0.35</u>	0.45	0.55
Accuracy	55.2	55.9	**56.9**	56.1	55.0
Skill use (%)	89.3	81.6	73.0	58.2	41.7
Cache capacity κ_c	5	<u>10</u>	20	50	
Accuracy	55.5	**56.9**	56.1	55.3	
Warm-up N_w	200	350	<u>500</u>	750	1000
Accuracy	54.8	55.6	**56.9**	56.2	55.5

The threshold δ trades utilization against selection precision: at $\delta = 0.15$ nearly 90% of rollouts receive an injected document, and the marginally relevant admissions cost 1.7 points, while $\delta = 0.55$ suppresses utilization to 42% and forfeits 1.9 points. The default configuration injects a document into 73% of rollouts, and ablations (a) and (d) of Table 4 move this rate to 28% and 88%. The default $\kappa_c = 10$ balances repertoire size against scoring cost, since halving the cache limits the candidates available to μ_θ, and expanding it to 50 dilutes the pool with low-utility entries while multiplying the scoring cost fivefold. Shortening warm-up to $N_w = 200$ yields the largest deficit of 2.1 points, because Phase II then starts from both a weak base policy and an underpopulated library, whereas $N_w = 1000$ delays co-evolution until half the training budget is spent. The default warm-up leaves the library with roughly 50 to 80 entries across both tiers before the policy over options activates.

F.2 Library Evolution across Training

Table 10 lists representative entries at steps 1,000 and 2,000. Entries concluded during warm-up are broad heuristics such as equation setup and parity checks, mid-training entries specialize to number theory and combinatorics, and late entries carry targeted techniques such as Vieta's formulas and inscribed-angle chasing that transfer to Olympiad-level problems in Omni-MATH. The specialization trajectory explains the larger out-of-distribution gains in Table 3, since the evolving library accumulates domain-specific reasoning patterns that the base policy does not develop under GRPO alone.

Table 10: Representative skill documents at two training stages show the progression from domain-level heuristics to targeted techniques.

Stage	**Skill name**	**Type**	**Key insight**
Step 1k	`modular_residue_analysis`	Number theory	Reduce expressions modulo n to constrain integer solutions before full computation
Step 1k	`generating_function_setup`	Combin.	Encode a counting sequence as polynomial coefficients and extract the target term
Step 2k	`vieta_root_reconstruction`	Algebra	Use Vieta's formulas to express symmetric functions of roots without solving the polynomial
Step 2k	`angle_chasing_with_inscribed`	Geometry	In cyclic quadrilaterals, the inscribed angle theorem converts arc relations to angle equations

Around the phase boundary at step $N_w + 1$, utilization rises from zero to roughly 40 to 50% within the first 10 to 20 steps of Phase II, as the policy over options begins sampling from a cache whose entries

were concluded during warm-up. Utilization then settles at 65 to 75% over the following 100 to 200 steps as the confidence gate filters low-relevance selections. The cumulative skill count and accuracy then rise together until step 1,500 in Figure 2(b), after which the count plateaus while accuracy continues to climb, so the LOAD and EVICT cycle refines the active pool instead of expanding it.

F.3 TOKEN AND COMPUTE OVERHEAD

Table 11 breaks down the per-step wall-clock time by component. Skill conclusion and the verification rollout contribute 10.1s of the 13.7s addition, log-likelihood scoring costs 1.8s with the cache capacity $\kappa_c = 10$ and documents capped at 128 tokens, and maintenance costs 0.6s. The overhead is therefore located in the two extra rollouts and does not grow with the library, since only cache entries enter the scoring pass.

Table 11: The table breaks down the per-step wall-clock time on 2×H200 with $G = 8$ by component.

Component	GRPO	ARISE
Solution rollouts	38.2s	39.1s
Reward and advantage	13.0s	13.3s
Skill selection		1.8s
Skill conclusion		5.2s
Verification rollout		4.9s
Library maintenance		0.6s
Total per step	51.2s	64.9s
Overhead		+26.8%

Table 12 compares average per-problem token counts on AIME 2024 across methods. The injected document adds an average of 66 prompt tokens per problem under the 220-character cap, and the response length of 2,923 tokens stays within 3% of the 2,847 of GRPO, so injection steers the strategy while leaving solution length nearly unchanged. The 4.5% total overhead buys a 3.1-point macro-average gain on Omni-MATH over GRPO in Table 3, whereas EvolveR pays 7.9% for 0.7 points, which places ARISE at the favorable end of Figure 2(a). DAPO and GSPO leave the prompt unchanged and incur their small overhead through longer responses.

Table 12: The table compares average per-problem prompt and response tokens on AIME 2024 for Qwen3-4B, with overhead measured relative to the GRPO total.

Method	**Prompt**	**Response**	**Total**	**Overhead**
GRPO	312	2,847	3,159	
DAPO	312	2,903	3,215	+1.8%
GSPO	312	2,871	3,183	+0.8%
EvolveR + GRPO	489	2,921	3,410	+7.9%
ARISE + GRPO	378	2,923	3,301	+4.5%

G ADDITIONAL COMPARISONS

G.1 ADAPTED SAGE BASELINE

SAGE (Wang et al., 2025) folds skills into GRPO training for multi-turn tool-use agents on AppWorld, where skills accumulate across a task chain through its Sequential Rollout and a skill-integrated reward complements the outcome reward, while the operations that decide which skill to invoke stay outside the policy gradient. SAGE lacks a native configuration for single-turn verifiable-answer reasoning, since the Sequential Rollout over a task chain lacks an analogue when each query is solved in one episode. We therefore constructed an adapted version that re-casts the skill-integrated reward and the library accumulation of SAGE onto single-turn mathematical rollouts under the training protocol of Appendix C.1, and we report it as a reference point, not as a faithful reproduction. Table 13 lists the result on Qwen3-4B-Instruct-2507 over three training seeds. The adapted SAGE lands with

the memory-augmented baselines, above GRPO and below the coupled design, consistent with the account that decoupling selection from the gradient leaves value on the table. SkillRL (Xia et al., 2026) targets embodied and search agents, and its released implementation builds skill retrieval and skill evolution on a recursive multi-turn interaction loop with environment feedback per step. Single-turn reasoning with one verifiable answer per query offers neither the intermediate observations nor the recursion that the loop requires, so the method does not adapt to our setting and is omitted from the comparison.

Table 13: The table reports adapted SAGE on Qwen3-4B in Pass@1 (%), with mean and standard deviation over three training seeds.

Method	AMC 23	AIME 24	AIME 25	Omni Avg.
GRPO	72.9±0.4	54.1±0.7	46.5±0.8	23.9±0.3
EvolveR + GRPO	73.6	54.8	46.3	24.6
SAGE (adapted) + GRPO	73.4±0.5	54.9±0.8	46.7±0.9	24.8±0.4
ARISE + GRPO	**75.6**±0.4	**56.9**±0.6	**48.3**±0.7	**27.0**±0.3

G.2 Decoupled Scoring Head

Ablation (f) of Table 4 replaces the shared-parameter scorer of Eq. (4) with a scoring head that carries its own parameters ϕ and is trained by REINFORCE on the same trajectory reward R^w. The remaining components of ARISE stay identical, including the library, the conclusion step, the verification rollout, and the maintenance rules. The variant preserves the option mechanism while removing the channel through which generation feedback reshapes selection under shared parameters. Table 14 reports the comparison on Qwen3-4B-Instruct-2507 over three training seeds, including AMC 2023, which the main-text ablation table omits for space.

Table 14: The table compares the decoupled scoring head with the shared-parameter design on Qwen3-4B in Pass@1 (%), with mean and standard deviation over three training seeds.

Configuration	AMC 23	AIME 24	AIME 25	Omni Avg.
GRPO (no library)	72.9±0.4	54.1±0.7	46.5±0.8	23.9±0.3
Decoupled scoring head	74.2±0.5	55.3±0.8	47.1±0.9	25.7±0.4
ARISE + GRPO (shared parameters)	**75.6**±0.4	**56.9**±0.6	**48.3**±0.7	**27.0**±0.3
Gain of shared over decoupled	+1.4	+1.6	+1.2	+1.3

The decoupled variant improves over GRPO on the four benchmarks, yet falls below the shared-parameter design by 1.2 to 1.6 points, beyond the seed standard deviation in the four columns. The variant matches random selection (b) of Table 4 to within 0.1 point on its three columns, consistent with Corollary B.1, under which the REINFORCE estimate of the head vanishes group by group and the library alone accounts for its gain over GRPO. The adapted SAGE result of Table 13 lies 0.4 to 0.9 points below the decoupled head, and the shared-parameter coupling separates from both decoupled designs.

G.3 Third Backbone

Table 15 extends the comparison to Qwen3-8B-Base (Team, 2025) under the protocol of Appendix C.1. The gain persists at the 8B scale and is again largest on the out-of-distribution Omni-MATH average, consistent with the two-model results in Table 3.

H Skill Transfer and Library Dynamics

H.1 Attribution on Unseen Queries

The skill quality level of §3.3 is computed on the query from whose ground-truth trace the skill was concluded, so a level could in principle reflect partial revelation of that query's solution path instead

Table 15: The table reports Pass@1 (%) on Qwen3-8B-Base under the identical training and evaluation protocol.

Method	AMC 23	AIME 24	Omni Avg.
GRPO	78.4	33.6	30.1
ARISE + GRPO	**80.6**$_{\uparrow 2.2}$	**35.9**$_{\uparrow 2.3}$	**32.8**$_{\uparrow 2.7}$

of a transferable pattern. To separate the two readings, we recomputed the level on unseen queries. For each of the ten cached skills at convergence on Qwen3-4B, with the ten highest-utility reservoir entries as a comparison group, we drew an unseen query of the same problem type from Omni-MATH, disjoint from the training queries. We then ran G unaided rollouts and G skill-conditioned rollouts and classified the outcome by the four-level taxonomy. Table 16 reports the counts, where positive attribution counts skills at level r_3 or r_4, meaning that the skill raises the probability of a correct solution above the unaided baseline on a query it has not seen.

Table 16: The table recomputes skill quality levels on unseen Omni-MATH queries of the same problem type for the converged cache and the top reservoir entries.

Skill source	Entries	Harmful r_1	Neutral r_2	Useful r_3	Critical r_4	Positive
Cache at convergence	10	1	1	5	3	8 of 10
Reservoir top-10	10	2	3	4	1	5 of 10

Among the ten cached skills, eight retain positive attribution on a query they were not concluded from. Leakage predicts that the marginal signal disappears once the source query is replaced, while transferable structure predicts that it persists, and the counts support the transfer reading. The cache group also transfers better than the reservoir group, consistent with the utility-gated maintenance rules promoting skills by transferable performance. The samples are small because the cache capacity fixes their size. Table 3 constrains leakage independently, since a library retained through leakage would gain least out of distribution, whereas the largest gain appears on Omni-MATH, whose 4,428 problems are disjoint from the training queries.

H.2 Displacement of Early Skills

Skills concluded early in Phase I come from a weak policy, and Table 5 records the consequence in the level composition: at the start of Phase I the neutral r_2 outcome accounts for 51.3% of concluded skills and the critical r_4 outcome for 25.9%. Such an entry enters the cache with a low initial utility, is rarely selected while stronger entries are available, and is displaced to the reservoir by EVICT and removed by DELETE once its utility falls below the reservoir's tenth percentile with zero usage. The maintenance rules are therefore gated by utility, not by recency of conclusion. Table 17 measures the dynamic directly on Qwen3-4B by tracking the entries concluded early in Phase I through Phase II.

Table 17: The table tracks the share and utility of the cache entries concluded early in Phase I through Phase II on Qwen3-4B alongside AIME 2024 accuracy.

Step	Cache share from early Phase I	Mean utility of early entries	AIME 24
500 (phase boundary)	100%	0.31	
800	62%	0.29	55.1
1,200	34%	0.27	56.2
2,000	11%	0.26	56.9

Early skills are steadily displaced by higher-utility entries concluded once the policy strengthens, their retained share falls to 11% by the end of training while accuracy rises monotonically, and early low-quality skills therefore neither accumulate nor slow convergence. The warm-up schedule is the second line of defense, since the policy over options stays disabled for the first N_w steps, and the sensitivity study of Appendix F.1 quantifies the cost of shortening it.

I CODE GENERATION SETUP

The code-generation experiment of §4.3 uses Qwen3-4B-Instruct-2507 as the backbone and unit-test pass as the binary verifiable reward, so $v(\tau, q)$ equals 1 when the generated program passes the full test suite attached to the problem. Training uses the DeepCoder corpus (Luo et al., 2025a) of approximately 24K competitive programming problems, each retained only after its official solution passes a suite of at least five unit tests, and the corpus is disjoint from HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). The official solution serves as the ground-truth trace τ_t^+ for skill conclusion. The skill-conclusion, library-maintenance, and warm-up procedure is identical to the mathematical setting, and the only change to the conclusion template of Figure 4 replaces the reasoning-strategy fields with algorithmic-pattern fields. The hyperparameters of Table 7 are transferred untouched, and training uses a reduced budget of 1,000 steps, half the 2,000 of the mathematical experiments, with the warm-up length $N_w = 500$ unchanged, since the purpose is to test cross-domain transfer, not absolute code performance. Results in Table 18 are averaged over three training seeds.

Table 18: The table reports Pass@1 (%) on code generation with Qwen3-4B, with mean and standard deviation over three training seeds.

Method	HumanEval	MBPP
GRPO	78.7$\pm$0.5	73.2$\pm$0.4
ARISE + GRPO	**80.5**$\pm$0.4$_{\uparrow 1.8}$	**75.6**$\pm$0.3$_{\uparrow 2.4}$

J LIMITATIONS

The evaluation covers mathematical reasoning and a code-generation extension, and multi-turn tool-use tasks remain a further step, since they require option termination and cross-step credit assignment that the single-option-per-query regime omits. Skill conclusion draws on reference traces at training time. The self-rollout variant (e) of Table 4 dispenses with the traces and retains 1.7 of the 2.8 AIME 2024 points over GRPO, so the traces sharpen the skills but are not a precondition for the gain. Training incurs a per-step overhead of 26.8% on the reported hardware, located in skill conclusion and the verification rollout.