Skill-Conditioned Gated Self-Distillation for LLM Reasoning

Jiazhen Huang[1], **Xiao Chen**[1], **Xiao Luo**[2], **Yong Dai**[3], **Senkang Hu**[4], **Yuzhi Zhao**[5]
[1]Tsinghua University, [2]University of Wisconsin-Madison, [3]X-Humanoid,
[4]City University of Hong Kong, [5]Huazhong University of Science and Technology
huangjiazhen1125@gmail.com, senkang.forest@my.cityu.edu.hk

Abstract

On-policy self-distillation (SD) improves LLM reasoning by using teacher-side privileged information (PI) to turn sparse verifier outcomes into dense token-level supervision. Existing methods usually assume trusted PI, such as reference answers or successful traces. We ask whether PI can instead come from an experience-derived skill bank, where retrieved skills are compact and reusable but may also be irrelevant or misleading. We propose Skill-Conditioned Gated Self-Distillation (SGSD), which formulates skill-based SD as teacher hypothesis validation rather than unconditional imitation. SGSD retrieves skill-mistake pairs, constructs a multi-teacher pool, and lets all skill-conditioned teachers score the same plain-prompt student rollout. The verifier validates each teacher's polarity: supporting a success or suppressing a failure gives positive supervision, while the opposite stance is reversed. A robust gated objective then distills informative teacher-student disagreements while suppressing uncertain or extreme signals. Experiments on multiple mathematical reasoning benchmarks show that SGSD consistently improves over GRPO and remains competitive with answer-conditioned OPSD under a weaker PI assumption. For example, on Qwen3-1.7B, SGSD outperforms GRPO by 6.2% and OPSD by 1.7% on AIME24, AIME25, and HMMT25.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central recipe for post-training large language models (LLMs) with reasoning capability. RLVR methods (e.g., GRPO (Shao et al., 2024)) sample solutions, verify the final answer, and improve the policy from environmental outcome feedback. This setting is attractive because it requires only a verifier, not token-level annotations or a larger teacher model. Its weakness is equally well known: the reward signal is sparse, delayed, and nearly uniform across the entire trajectory. On-policy distillation (OPD) (Agarwal et al., 2024) addresses this limitation by letting a stronger teacher provide dense token-level supervision on the student's sampled rollouts. On-policy self-distillation (SD)[1] goes one step further: the same model acts as teacher and student under different contexts, removing the need for a separate teacher model (Zhao et al., 2026; Cui et al., 2026). Therefore, SD promises the on-policy nature of RLVR together with richer credit assignment.

The effectiveness of SD depends on the teacher-side privileged context information (PI). Prior work usually gives the teacher a reference solution, a successful trace, textual feedback, or demonstrations (Zhao et al., 2026; Yang et al., 2026; Hübotter et al., 2026; Shenfeld et al., 2026). In mathematical reasoning, this raises a natural question: can PI instead come from a structured *skill bank*, as in prior work for agentic tasks (Wang et al., 2026)? Skills (Anthropic, 2024) are a common abstraction in agentic learning and memory-like systems: previous trajectories are compressed into reusable natural-language principles, tactics, and mistake warnings (Shinn et al., 2023; Zhao et al., 2024; Xia et al., 2026). Compared with full traces or reference answers, skills are more compact and potentially more reusable across problems, making them appealing for mathematical reasoning in a weaker supervision regime. Here, training may have only verifier feedback rather than a ground-truth answer.

However, skills introduce a harder reliability problem than the privileged contexts used in standard SD. A retrieved skill can be relevant, partially relevant, stale, or simply wrong for the current problem. More importantly, even a reasonable skill should not be copied unconditionally. For example: supporting a successful rollout is helpful, but

[1]To distinguish from another representative method OPSD, we uniformly use the abbreviation "SD" throughout this paper.

Table 1: **Comparison of representative post-training paradigms for LLM reasoning.** SGSD keeps on-policy dense self-supervision under a weaker PI assumption, using environment (verifier) outcomes and skill-conditioned teacher signals to decide the update direction.

	Sampling	Signal	Teacher	PI Assumption	Direction Anchoring
Distillation	× off-policy	✓ dense	× external	—	× teacher
On-policy distillation	✓ on-policy	✓ dense	× external	—	× teacher
RLVR (e.g., GRPO)	✓ on-policy	× sparse	—	—	× environment
OPSD	✓ on-policy	✓ dense	✓ self	× strong	× teacher
SGSD (ours)	✓ on-policy	✓ dense	✓ self	✓ weak	✓ environment & teacher

supporting a failed rollout is harmful; suppressing an incorrect trajectory is useful, but suppressing a correct one is not. This leads to a central question: *can experience-derived skills serve as training-only privileged information without assuming that every retrieved skill is trusted?*

Answering this question requires solving three challenges: **(1)** First, the privileged information is no longer a single, trusted source but a set of retrieved skill hypotheses with uncertain relevance. **(2)** Second, the value of a skill is outcome-dependent: the same teacher support can be beneficial or misleading depending on whether the sampled rollout succeeds. **(3)** Third, multiple skill-conditioned teachers can produce noisy or conflicting token-level signals, so the method must extract useful disagreement without being dominated by prompt artifacts or extreme logits.

Therefore, we propose **Skill-Conditioned Gated Self-Distillation (SGSD)**. SGSD maintains a structured bank of general skills and common mistake patterns distilled from past trajectories. For each problem, it retrieves relevant skills and mistakes, pairs them into a skill-conditioned multi-teacher pool, and lets all teachers score the same student rollout while the student itself sees only the plain problem. SGSD then combines each teacher's support with the rollout outcome to infer its polarity: a teacher is helpful when it supports a success or suppresses a failure, and harmful otherwise. A robust gated objective finally learns from informative teacher-student disagreements while down-weighting ambiguous or extreme signals. These designs enable SGSD to achieve comparable performances even under a weaker PI assumption. Moreover, in contrast to prior distillation methods whose update direction is controlled by a teacher alone, SGSD derives the direction jointly from the environment outcome and the skill-conditioned teacher support. Tab. 1 summarizes the differences between SGSD and related paradigms.

Our contributions can be concluded as follows:

- We introduce skills and self-distillation to mathematical reasoning with verifier feedback, and formulate it as a teacher hypothesis validation problem rather than assuming retrieved skills should be imitated by default.
- We propose SGSD, a multi-teacher, skill-conditioned self-distillation framework that infers teacher polarity from the alignment between teacher support and rollout outcome, then applies a robust gated objective.
- We provide empirical results on several mathematical reasoning benchmarks, showing that our SGSD improves over GRPO and remains competitive with SD baselines. For example, on Qwen3-1.7B, SGSD improves the average score across AIME24, AIME25, and HMMT25 from 37.4 to 43.7.

2 Our proposed SGSD

Let $x \sim \mathcal{D}$ denote a problem and y a sampled reasoning trajectory. SGSD uses one policy model π_θ in two roles. The student is the plain-prompt policy $p_S(\cdot \mid x) = \pi_\theta(\cdot \mid x)$, which samples

$$y \sim p_S(\cdot \mid x). \tag{1}$$

After generation, an answer verifier gives a scalar outcome $r \in \{-1, 1\}$. The student is never given retrieved skills during rollout or evaluation. Skills are only injected as teacher-side PI during training.

2.1 Structured Skill Bank Construction

SGSD uses a structured skill bank $\mathcal{B} = \{\mathcal{G}, \mathcal{E}\}$ rather than raw historical trajectories[2]. $\mathcal{G}$ contains *general skills*, which encode reusable reasoning principles with a short title and conditions for use. $\mathcal{E}$ contains *common mistake patterns*, which describe an error, why it occurs, and how to avoid it. Detailed schemas and prompt templates are given in App. C and App. D, respectively.

The bank is initialized before training by a cold-start extraction process. We first sample a subset

[2]In this paper, a skill means an experience-derived natural-language principle, not an external tool or executable function.

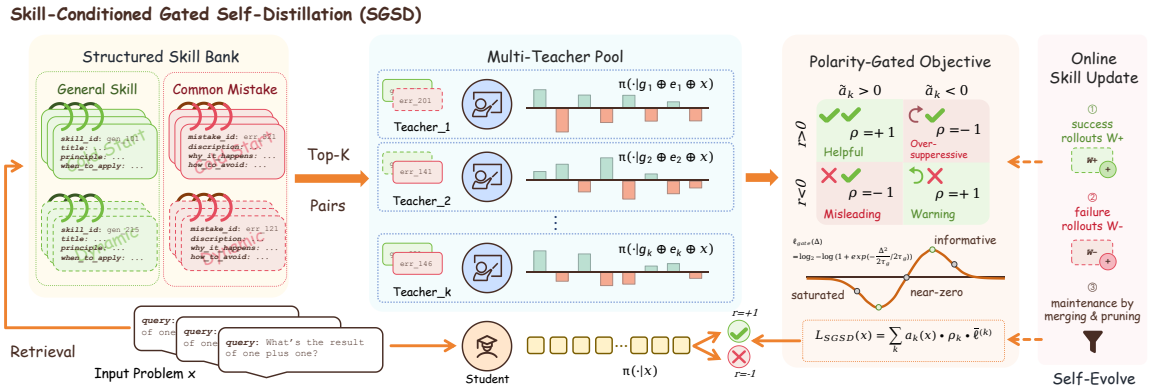

Figure 1: **Overview of SGSD.** The student samples a rollout from the plain problem, while retrieved skill–mistake pairs instantiate a pool of skill-conditioned teachers. Multiple teachers score the same rollout, the verifier outcome validates their polarity, and a gated distillation objective turns teacher-student gaps into dense supervision.

of training problems as the cold-start seed set and generate solution traces for them with the initial policy. These traces are then converted into structured records. Specifically, successful traces are summarized into candidate general skills, failed traces are summarized into candidate mistake patterns, and redundant candidates are merged into a compact bank. This follows the same spirit as skill-library construction in agent learning (Xia et al., 2026): preserve reusable experience while removing noisy trajectory details. App. D.2 gives the concrete construction procedure.

The central unit used by SGSD is a skill pair (g_k, e_k), where $g_k \in \mathcal{G}$ provides positive guidance about what to do and $e_k \in \mathcal{E}$ provides negative guidance about what to avoid. Pairing the two creates a balanced teacher context instead of appending a large unstructured memory block. During training, each retrieved pair becomes a separate teacher-side PI block, and all resulting teachers score the same student rollout.

2.2 Skill-Conditioned Multi-Teacher Pool

For a new problem x, SGSD performs semantic-similarity retrieval over the skill bank, obtaining:

$$\mathcal{G}(x) = \{g_1, \ldots, g_{K_x}\}, \quad \mathcal{E}(x) = \{e_1, \ldots, e_{K_x}\}. \tag{2}$$

The k-th retrieved skill and mistake are concatenated with the problem to form the corresponding teacher-side PI:

$$c_k(x) = g_k \oplus e_k \oplus x, \tag{3}$$

where $\oplus$ denotes prompt concatenation. These contexts define a pool of K_x skill-conditioned teachers. The k-th teacher distribution is

$$p_T^{(k)}(y_t \mid x, y_{<t}) = \pi_{\bar{\theta}}(y_t \mid c_k(x), y_{<t}), \tag{4}$$

where $\bar{\theta}$ is the stop-gradient copy of the current policy parameters. Thus, all teachers share the same policy model as the student, but differ in the skill-conditioned PI they observe. The student distribution remains

$$p_S(y_t \mid x, y_{<t}) = \pi_\theta(y_t \mid x, y_{<t}). \tag{5}$$

SGSD assigns each teacher a relevance weight $\alpha_k(x)$ from the semantic retrieval scores. Let $s_k^{(g)}$ and $s_k^{(e)}$ be the similarity scores of the paired skill and mistake; we use the average score $s_k(x) = (s_k^{(g)}(x) + s_k^{(e)}(x))/2$ and normalize

$$\alpha_k(x) = \frac{\exp(s_k(x))}{\sum_{j=1}^{K_x} \exp(s_j(x))}. \tag{6}$$

2.3 Outcome-Validated Teacher Polarity

The purpose of SGSD is not to imitate every teacher induced by retrieved skills, but to decide which teacher stances agree with the observed outcome and thus deserve to be distilled.

For each generated token y_t, SGSD computes the teacher-student log-probability gap:

$$\Delta_t^{(k)} = \log p_T^{(k)}(y_t \mid x, y_{<t}) - \log p_S(y_t \mid x, y_{<t}). \tag{7}$$

This gap is the token-level credit that appears in reverse-KL-style policy-gradient form. Positive $\Delta_t^{(k)}$ means teacher k assigns higher probability to the sampled token than the student does and thus supports the token, while negative $\Delta_t^{(k)}$ means the teacher suppresses the sampled token.

Next, define the plain teacher-level support score

$$a_k = \frac{1}{T} \sum_{t=1}^{T} \Delta_t^{(k)}. \tag{8}$$

Table 2: **Decision rule of teacher polarity.** SGSD distills a skill-conditioned teacher when its polarity agrees with the verifier outcome, and reverses it otherwise.

Decision	Rule	Case	Interpretation
Distill	$\mathrm{sign}(r) = \mathrm{sign}(a_k)$	$r > 0,\ a_k > 0$	Helpful teacher
		$r < 0,\ a_k < 0$	Warning teacher
Reverse	$\mathrm{sign}(r) \neq \mathrm{sign}(a_k)$	$r > 0,\ a_k < 0$	Over-suppressive teacher
		$r < 0,\ a_k > 0$	Misleading teacher

The direction of a_k summarizes whether teacher k supports or suppresses the sampled rollout over the whole trajectory. The outcome then validates this stance. Define

$$\mathrm{sgn}_{\mathrm{out}}(r) = \begin{cases} 1, & r > 0, \\ -1, & r < 0, \end{cases} \tag{9}$$

the plain polarity is

$$\rho_k^{\mathrm{plain}} = \mathrm{sgn}_{\mathrm{out}}(r)\,\mathrm{sgn}(a_k). \tag{10}$$

Here, $\mathrm{sgn}(0) = 0$. Therefore, a teacher is useful when it supports a successful rollout or suppresses a failed rollout, while it is misleading when it supports a failure or suppresses a success. Tab. 2 summarizes the decision rule.

2.4 Robust Support Estimation

The plain support score a_k can be distorted by formatting tokens, prompt artifacts, or a few extreme token gaps. SGSD therefore estimates a robust support score $\widetilde{a}_k$ before deciding the final polarity. A token mask $m_t \in \{0, 1\}$ first selects effective reasoning positions and removes special tokens, chat-template tokens, thinking delimiters, and pure-formatting tokens from aggregation. For support estimation, each gap is then clipped as

$$\widetilde{\Delta}_t^{(k)} = \mathrm{clip}(\Delta_t^{(k)}, -c_\Delta, c_\Delta), \tag{11}$$

where $c_\Delta > 0$ is the clipping bound. The robust support score is

$$\widetilde{a}_k = \frac{\sum_{t=1}^{T} m_t \widetilde{\Delta}_t^{(k)}}{\sum_{t=1}^{T} m_t + \epsilon}, \tag{12}$$

where $\epsilon > 0$ prevents division by zero. Furthermore, the final polarity uses a confidence threshold $\epsilon_a \geq 0$ to allow uncertain teachers to be ignored:

$$\rho_k = \begin{cases} \mathrm{sgn}_{\mathrm{out}}(r)\,\mathrm{sgn}(\widetilde{a}_k), & |\widetilde{a}_k| > \epsilon_a, \\ 0, & |\widetilde{a}_k| \leq \epsilon_a. \end{cases} \tag{13}$$

Overall, masking preserves the effective reasoning tokens, clipping prevents outlier gaps from flipping the teacher-level stance, and thresholding creates a neutral zone for uncertain evidence.

2.5 Gated Distillation Objective

After polarity decides whether a teacher should be distilled, reversed, or ignored for the current rollout, the remaining question is how strongly to learn from each teacher-student disagreement. Directly optimizing a sampled-token reverse-KL-style gap can overreact to very large discrepancies: extreme positive or negative gaps may dominate the optimization, while near-zero gaps carry little information. SGSD therefore uses a bounded gated loss:

$$\ell_{\mathrm{gate}}(\Delta) = \log 2 - \log\left(1 + \exp\left(-\frac{\Delta^2}{2\tau_g}\right)\right), \tag{14}$$

where $\tau_g > 0$ controls the width of the informative region, and the $\log 2$ constant normalizes the loss to be zero at $\Delta = 0$. The loss value is near zero when teacher and student agree, grows for moderate disagreement, and saturates for extreme mismatch. And, the induced update remains bounded rather than being dominated by outlier gaps. Section 3 shows that its gradient yields a bounded policy-gradient-style correction and can be viewed as a sampled-token, polarity-aware approximation to local reverse-KL behavior.

For each skill-conditioned teacher, the gated loss is

$$\bar{\ell}^{(k)} = \frac{\sum_{t=1}^{T} m_t \ell_{\mathrm{gate}}(\Delta_t^{(k)})}{\sum_{t=1}^{T} m_t + \epsilon}. \tag{15}$$

The overall distillation objective is then given by:

$$\mathcal{L}_{\mathrm{SGSD}}(x) = \sum_{k=1}^{K_x} \alpha_k(x)\rho_k \bar{\ell}^{(k)}. \tag{16}$$

Since the teachers are stop-gradient, ρ_k controls the direction of the update: helpful teachers distill their stance, misleading teachers reverse it, and uncertain teachers with $\rho_k = 0$ contribute nothing.

2.6 Online Skill Update and Maintenance

SGSD can update its skill bank during training, so that retrieved skills can reflect the current policy's observed successes and failures. At fixed update intervals, recent student rollouts are collected into a window $\mathcal{W}$ with verifier outcomes. When recent success rate is already high, the update is skipped. Otherwise, the window is split into successful traces $\mathcal{W}^+$ and failed traces $\mathcal{W}^-$. Successful traces are summarized into new general-skill candidates, while failed traces are summarized into new common-mistake candidates. New candidates are then merged with existing bank entries, written back as dynamic entries, and pruned under a

Algorithm 1 Overall Procedure of SGSD

Require: Training set $\mathcal{D}$, policy model π_θ, skill bank $\mathcal{B} = \{\mathcal{G}, \mathcal{E}\}$, retrieved pair count K

for each update **do**
  Sample a minibatch $\mathcal{X} \subset \mathcal{D}$
  for each problem $x \in \mathcal{X}$ **do**
    Retrieve K ranked skill–mistake pairs $\{(g_k, e_k)\}_{k=1}^{K_x}$ from $\mathcal{B}$
    Compute teacher weights $\{\alpha_k(x)\}_{k=1}^{K_x}$ from retrieval scores
    Form skill-conditioned teacher contexts $c_k(x) = g_k \oplus e_k \oplus x$
    Sample a plain-prompt student rollout $y \sim p_S(\cdot \mid x)$ and evaluate outcome r
    for each teacher context $c_k(x)$ **do**
      Score the fixed rollout with the stop-gradient teacher: $p_T^{(k)}(y_t \mid x, y_{<t}) = \pi_{\bar{\theta}}(y_t \mid c_k(x), y_{<t})$
      Compute token gaps $\Delta_t^{(k)}$ and robust support $\widetilde{a}_k$
      Infer polarity ρ_k from $\widetilde{a}_k$ and r
      Compute the gated teacher loss $\bar{\ell}^{(k)}$
    end for
    Accumulate the per-example objective: $\mathcal{L}_{\mathrm{SGSD}}(x) = \sum_{k=1}^{K_x} \alpha_k(x)\rho_k\bar{\ell}^{(k)}$
  end for
  Update θ with the minibatch SGSD objective
  Update $\mathcal{B}$ from recent successful and failed rollouts
end for

capacity limit to keep the bank compact. App. D.4 gives the detailed update procedure, and Alg. 1 summarizes the overall procedure of SGSD.

3 Theoretical Analysis

This section analyzes why the gated distillation objective works. The key is its derivative

$$g_\tau(\Delta) = \frac{\partial \ell_{\mathrm{gate}}(\Delta)}{\partial \Delta} = \frac{\Delta}{\tau_g\left(1 + \exp\left(\Delta^2/(2\tau_g)\right)\right)}, \quad (17)$$

because g_τ directly determines how each teacher-student gap enters the policy gradient. Detailed derivations are deferred to Apps. E and E.5, with a visualization in App. E.2.

Policy-gradient form. Since teacher distributions are stop-gradient targets, we have

$$\nabla_\theta \Delta_t^{(k)} = -\nabla_\theta \log p_S(y_t \mid x, y_{<t}). \quad (18)$$

Let $Z = \sum_{t=1}^{T} m_t + \epsilon$. Substituting the derivative of ℓ_{gate} into Eq. (16) gives

$$-\nabla_\theta \mathcal{L}_{\mathrm{SGSD}}(x) = \sum_{k=1}^{K_x}\sum_{t=1}^{T} \alpha_k(x)\rho_k \frac{m_t}{Z} g_\tau(\Delta_t^{(k)}) \cdot \nabla_\theta \log p_S(y_t \mid x, y_{<t}). \quad (19)$$

Thus SGSD is a policy-gradient-style update with dense token-level coefficient

$$W_t^{(k)} = \alpha_k(x)\rho_k \frac{m_t}{Z} g_\tau(\Delta_t^{(k)}). \quad (20)$$

In this sense, $W_t^{(k)}$ acts as an advantage-like dense credit. The effectiveness of the loss reduces to whether g_τ induces the right gradient shape. Specifically, this gradient has three properties as follows.

Sign consistency. We have

$$\mathrm{sign}(g_\tau(\Delta)) = \mathrm{sign}(\Delta) \quad (\Delta \neq 0). \quad (21)$$

Therefore, after applying ρ_k, helpful teachers reinforce their supporting tokens and misleading teachers reverse them.

Bounded informative region. We introduce the following proposition:

Proposition 3.1. *For every $\Delta \in \mathbb{R}$, we have*

$$|g_\tau(\Delta)| \leq \frac{1}{\sqrt{e\,\tau_g}}, \quad (22)$$

Furthermore,

$$g_\tau(\Delta) = \frac{\Delta}{2\tau_g} + O\left(\frac{\Delta^3}{\tau_g^2}\right) \quad \text{as } \Delta \to 0, \quad (23)$$

while

$$g_\tau(\Delta) = \frac{\Delta}{\tau_g}\exp\left(-\frac{\Delta^2}{2\tau_g}\right)(1 + o(1)) \quad \text{as } |\Delta| \to \infty. \quad (24)$$

This proposition captures the intended behavior of the gated loss: (1) Near zero, the update vanishes linearly, so negligible teacher-student disagreements do not inject noise. (2) For very large gaps, the exponential factor suppresses extreme mismatches instead of letting them dominate optimization. (3) Between these two regimes, the update is concentrated on medium gaps of order $\sqrt{\tau_g}$, which is exactly the informative region where the teacher disagrees enough to matter but not so much that the signal is likely driven by outliers.

Local reverse-KL approximation. The same derivative also explains why the gated loss behaves like a reasonable distillation objective in the informative region. When teacher and student are locally close, reverse KL admits the second-order expansion

$$D_{\mathrm{KL}}(p_T \,\|\, p_S) = \frac{1}{2}\sum_v p_{T,v}\Delta_v^2 + O(\|\Delta\|^3), \quad (25)$$

where $\Delta_v = \log p_T(v \mid h_t) - \log p_S(v \mid h_t)$. Hence, the local reverse-KL gradient is linear in the log-probability gap. Our gated objective has the matching local form

$$g_\tau(\Delta) = \frac{\Delta}{2\tau_g} + O\left(\frac{\Delta^3}{\tau_g^2}\right), \quad (26)$$

so in the locally close regime, SGSD behaves like a sampled-token, polarity-aware reverse-KL correction up to a constant scale. Outside that regime, the gate deliberately departs from reverse KL by exponentially damping extreme gaps.

4 Experiments

4.1 Experimental Setup

Models and datasets. We use Qwen3-1.7B, Qwen3-4B, and Qwen3-8B as the base models (Yang et al., 2025). We train on the English subset of DAPO-Math-17K (Yu et al., 2026), and evaluate out of domain on three challenging competition-style mathematical reasoning benchmarks: AIME24, AIME25, and HMMT25.

Baselines and evaluation. We compare our method against **(1) GRPO** (Shao et al., 2024), a standard RLVR baseline that optimizes group-relative rewards from sampled rollouts, and **(2) OPSD** (Zhao et al., 2026), a representative SD baseline that uses the reference solution as teacher-side PI for dense token-level self-supervision. Different from the standard OPSD implementation, we provide the teacher with the ground-truth answer rather than a full distillation trajectory. We also include several skill-augmented variants: **(1) Base+Skill** retrieves skills at inference; **(2) GRPO+Skill** injects retrieved skills into training rollouts but evaluates with the plain problem prompt; and **(3) OPSD+Skill** exposes both teacher and student to skills during training and also evaluates without skills. We train the base models for 200 steps, checkpoint every 25 steps, and report the highest avg@12 accuracy among recorded checkpoints.

Implementation details. For each model, we randomly select 256 in-domain problems from the training set to build the cold-start skill bank. The teacher policy is synchronized with the student during training. Following OPSD, we disable thinking mode for the student and enable thinking mode for the teacher. We use Qwen3-Embedding-0.6B as the retriever to retrieve the top-8 skill pairs, and compute full-vocabulary distillation. The skill bank is updated every 25 steps. All experiments are conducted on eight NVIDIA A800 GPUs with LoRA adaptation (Hu et al., 2021), TRL framework (von Werra et al., 2020). For more experimental details, please refer to App. F.

4.2 Main Results

Skill-conditioned SD improves reasoning under weaker PI. Tab. 3 shows the main comparison. GRPO brings only limited gains, which is consistent with the sparsity of final-answer, trajectory-level rewards. OPSD and SGSD turn sparse rewards into dense token-level supervision, yielding stronger improvements across model scales. Surprisingly, SGSD performs competitively with OPSD, and even achieves better performance on smaller models. For example, on Qwen3-1.7B, SGSD improves the average score from 37.4% to 43.7%, outperforming GRPO by 6.2% and OPSD by 1.7%. This is achieved under a weaker PI assumption: OPSD uses a trusted reference answer as PI, whereas SGSD only assumes a verifier and a skill pair that may be misleading. Although SGSD still underperforms OPSD on Qwen3-8B, the results already suggest that outcome-validated skill-conditioned teachers can also approach answer-conditioned teachers for reasoning.

Skill injection alone is not sufficient. The skill-augmented variants do not perform well in Tab. 3, showing that the gains do not come from simply exposing the model to more skills. Base+Skill can underperform the plain base model, likely because a single retrieval step does not always return skills that are useful for the current reasoning problem. GRPO+Skill and OPSD+Skill show a similar issue: when skills are directly visible during training, the model may rely on them to solve the problem, rather than internalizing them to reason without skills. This issue is even clearer for OPSD+Skill, where teacher and student observe the same skills, leaving little teacher-student contrast for distillation. SGSD instead keeps skills on the teacher side only, using a multi-teacher pool so that skill-conditioned teachers can provide relevant guidance for the student to internalize.

4.3 Training Dynamics

Fig. 2 illustrates the training dynamics of on Qwen3-1.7B over 200 update steps. As shown, GRPO often stays below the base model, reflecting the weak learning signal from sparse outcome rewards. OPSD improves rapidly in the early stage, but gradually degrades after the mid-training checkpoints. In contrast, SGSD shows a more stable improvement pattern, reaching 42.3% average performance at step 100 and keeps a clear advantage over OPSD in the later stage. By validating teacher

Table 3: **Main comparison results on mathematical reasoning benchmarks.** The best and second-best results are highlighted in **bold** and underlined, respectively. We report the avg@12 accuracy for the best checkpoint on AIME24, AIME25, and HMMT25. Skill-conditioned SGSD improves over GRPO, and remains competitive with answer-conditioned OPSD under a weaker PI assumption.

Model	Method	AIME24	AIME25	HMMT25	Avg.
Qwen3-1.7B	Base	51.1	36.9	24.2	37.4
	Base+Skill	49.2	38.3	20.3	35.9
	GRPO	50.0	36.7	25.8	37.5
	GRPO+Skill	52.2	38.9	25.8	39.0
	OPSD	55.8	43.3	26.9	42.0
	OPSD+Skill	54.4	40.3	27.2	40.6
	SGSD	**57.8**	**43.6**	**29.7**	**43.7**
Qwen3-4B	Base	73.9	69.7	43.3	62.3
	Base+Skill	71.9	65.6	42.2	59.9
	GRPO	**76.1**	66.7	45.3	62.7
	GRPO+Skill	73.9	65.3	44.7	61.3
	OPSD	75.3	69.2	46.1	63.5
	OPSD+Skill	75.3	66.7	44.7	62.2
	SGSD	75.3	**70.8**	**46.7**	**64.3**
Qwen3-8B	Base	75.3	66.1	45.0	62.1
	Base+Skill	78.3	68.1	40.8	62.4
	GRPO	**79.2**	69.7	46.1	65.0
	GRPO+Skill	78.3	66.9	45.0	63.4
	OPSD	**79.2**	**73.1**	**48.1**	**66.8**
	OPSD+Skill	77.8	65.3	43.9	62.3
	SGSD	78.9	70.6	46.9	65.5

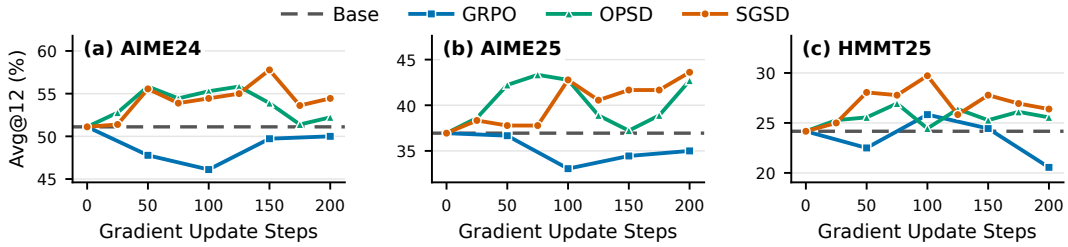

Figure 2: **Training dynamics on Qwen3-1.7B.** GRPO shows limited improvement under sparse outcome rewards, while SGSD maintains more stable gains than OPSD in the later training stage.

support with verifier outcomes and suppressing uncertain or extreme updates, SGSD keeps useful dense supervision available as both the policy and the skill bank evolve online.

4.4 Ablation Studies

We ablate the main components of SGSD on Qwen3-1.7B model, AIME24 dataset.

Robust support estimation improves training stability. As shown in Tab. 4, removing either token masking, clipping, or thresholding weakens model performance, and this phenomenon becomes more obvious in the later training stage. Removing all three even falls below the base model at step 100. Together, these components enable robust and stable distillation.

Teacher polarity and diversity are essential. Fig. 4 shows that removing polarity yields a short-term gain but eventually leads to collapse. Without polarity, all teachers are distilled in the same direction, introducing misleading supervision. Moreover, the variant without the multi-teacher pool consistently underperforms SGSD. This highlights the importance of teacher diversity: a single retrieved skill–mistake pair may be irrelevant or only partially useful for the problem, while the multi-teacher pool provides multiple hypotheses and a better chance to distill relevant guidance.

4.5 Further Discussions

We further study several design choices to understand when skill-conditioned SD remains reliable.

Full-vocabulary vs. Top-K support. Fig. 3(a) compares the default full-vocabulary distillation[3]

[3]Note that the objective remains defined on the sampled tokens through gap $\Delta_t^{(k)}$. The term "full-vocabulary" indicates that $p_T(\cdot)$ and $p_S(\cdot)$ are normalized over the full vocabulary and then used to compute $\Delta_t^{(k)}$ and support score a_k.

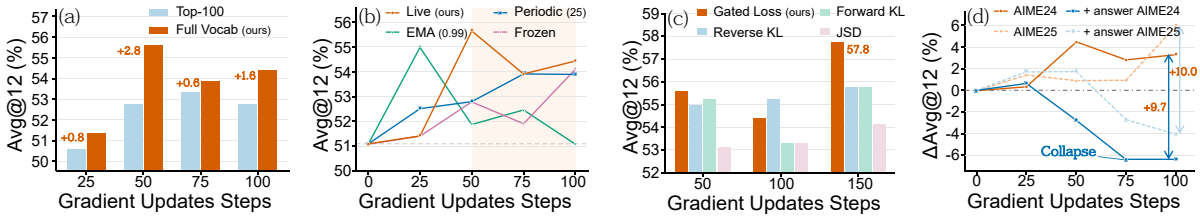

Figure 3: **Further discussions on Qwen3-1.7B.** (a) Full-vocabulary distillation consistently outperforms Top-100 support distillation on AIME24. (b) The live (synchronized) teacher used by SGSD reaches the strongest later checkpoints, while alternative update strategies either lag or fluctuate. (c) The gated objective obtains the highest peak among divergence-style losses on AIME24. (d) Compared with vanilla SGSD, adding OPSD-style answer PI to skill-conditioned teachers causes a large performance drop and thus collapse on both AIME24 and AIME25.

Table 4: **Robust support estimation ablations on Qwen3-1.7B, AIME24.** Arrows indicate changes relative to the Avg@12 accuracy of base model.

Method	Step 50	Step 100
Base	51.1	
SGSD	**55.6** ↑4.5	**54.4** ↑3.3
w/o token mask	53.1 ↑2.0	54.2 ↑3.1
w/o clipping	53.6 ↑2.5	52.5 ↑1.4
w/o threshold	54.4 ↑3.3	51.4 ↑0.3
w/o all three	54.2 ↑3.1	48.1 ↓3.0

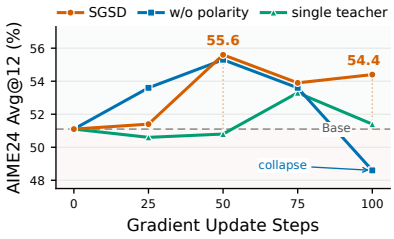

Figure 4: **Core ablations on Qwen3-1.7B.** Removing polarity causes late-stage collapse, while replacing the teacher pool with a single teacher remains below SGSD.

with a top-100 support approximation. For top-k support, we re-normalize both distributions on the teacher's local top-k support along with the sampled token, compute the resulting truncated local reverse-KL term (Fu et al., 2026), and apply the same gated objective. As shown, full-vocabulary distillation performs better at every checkpoint. This suggests that tail probabilities still help calibrate teacher-student log-probability gaps, and truncating to top-k can distort support estimation and weaken polarity-aware supervision.

Teacher policy update. Fig. 3(b) studies alternative teacher policy update strategies, including frozen, periodic, and EMA teachers. Synchronizing the teacher with the student is an aggressive choice, since both policies may confidently drift together during self-distillation. Nevertheless, the live teacher used by SGSD surprisingly performs well, reaching the highest peak accuracy and remain stable. This suggests that outcome-derived polarity and robust support estimation can stabilize training and make fresh teacher signals usable.

Divergence-style distillation objective. Fig. 3(c) compares our gated loss with several standard divergence-style objectives. Reverse KL is stable and competitive, but the gated loss obtains the highest peak, while forward KL and JSD are generally weaker. This matches the intended role of the gate: it preserves local reverse-KL-style correction for informative gaps while suppressing negligible disagreements and extreme mismatches.

Adding answer PI to SGSD. Fig. 3(d) adds the OPSD-style answer PI to the skill-conditioned teacher context. We observe that although this variant slightly improves at early stage, it soon consistently degrades and finally collapses on both AIME24 and AIME25, leaving SGSD ahead by 9.7% and 10.0% at step 100. One possible reason is that answer PI and polarity-controlled skill PI impose different information structures: when some teachers are intentionally reversed, attaching the same answer to every teacher may make the direction signal inconsistent. Thus, SGSD benefits from reusable skill PI, rather than simply giving the teacher richer privileged information.

5 Related Work

Outcome-driven reasoning optimization. Large language model reasoning is commonly improved with preference, reward, or verifiable outcome signals. RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023) optimize behavior from human or preference supervision, while RLVR methods (Chen et al., 2026; Ma et al., 2026; Liu et al., 2026b) optimize answers that can be checked automatically, with GRPO-style training (Shao et al., 2024; Zhai et al., 2026) widely used for mathematical reasoning. Recent work further studies how

to make outcome supervision more informative through self-rewarding signals and turn-level credit assignment (Yuan et al., 2024; Hu et al., 2026a), inference-time budget control, decoding-time task adaptation, retrieval rewards, and noise-tolerant training schedules (Fang et al., 2026; Hu et al., 2026c,b; Xu et al., 2025). These methods face the same limitation: rewards are sparse, delayed and nearly uniform over the trajectories.

On-policy distillation and self-distillation. Knowledge distillation transfers behavior from a teacher to a student (Hinton et al., 2015). Traditional distillation is usually performed on a fixed dataset, which creates a mismatch between the trajectories seen during training and those produced at inference time. On-policy distillation (OPD) (Agarwal et al., 2024) addresses this issue by letting the student sample its own trajectories while a stronger teacher provides dense token-level supervision on those trajectories. For reasoning models, this makes distillation more compatible with post-training settings that need fine-grained credit assignment. Recent on-policy self-distillation (SD) methods go one step further by removing the external teacher: the same model acts as both teacher and student, while the teacher is conditioned on privileged context such as reference solutions, revised attempts, textual feedback, or context-conditioned guidance (Zhao et al., 2026; Hübotter et al., 2026; Shenfeld et al., 2026; Ye et al., 2026; Chen et al., 2025). Other extensions study self-revision, unified distillation objectives, or sample routing under the same goal of converting sparse outcomes into dense token-level updates (He et al., 2026; Jin et al., 2026; Li et al., 2026a).

Experience, skills, and agent memory. A complementary line of work represents prior behavior as reusable natural-language experience. Chain-of-thought (CoT) (Wei et al., 2022) and STaR (Zelikman et al., 2022) reuse reasoning traces for self-improvement. Reflexion, Voyager, ExpeL (Shinn et al., 2023; Wang et al., 2023; Zhao et al., 2024), and collaborative LLM-agent systems (Hu et al., 2024, 2025; Huang et al., 2026) study how language-model agents reuse experience or coordinate decisions across tasks. Recent skill-based methods make this abstraction more explicit by distilling trajectories into reusable skills (Ni et al., 2026) or evolving skill libraries over time (Wang et al., 2025; Zhang et al., 2026b; Xia et al., 2026; Zhang et al., 2026a; Guo et al., 2026). SGSD does not inject retrieved skills into the student prompt at evaluation. Instead, skills are used as uncertain teacher-side hypotheses during training.

6 Conclusion

We introduced SGSD, a skill-conditioned self-distillation framework for mathematical reasoning with verifier feedback. Rather than assuming that retrieved skills should always be trusted, SGSD treats skill–mistake pairs as teacher hypotheses, validates their polarity with rollout outcomes, and learns from informative teacher-student disagreements through a robust gated objective. Experiments on competition-style mathematical reasoning benchmarks show that SGSD improves over GRPO and simple skill-injection baselines, while remaining competitive with answer-conditioned OPSD under a weaker PI assumption. These results suggest that experience-derived skills can provide useful dense supervision when their direction is validated by on-policy evidence.

Limitations

SGSD is evaluated mainly on mathematical reasoning tasks with automatic answer verification. This setting is well suited to our goal of studying outcome-validated skill-conditioned teachers, while extensions to open-ended generation may require a calibrated judge or another reliable outcome signal. Our skill bank uses a simple cold-start construction and rank-wise pairing between general skills and mistake patterns; more adaptive retrieval, pairing, or weighting strategies may further improve the teacher pool. The main experiments report the best checkpoint among recorded training steps for comparability with recent self-distillation work, so future studies could complement this protocol with held-out model selection. Finally, most ablations are conducted on Qwen3-1.7B to control computational cost, leaving a fuller sweep across larger models and additional hyperparameters for future work.

Ethical Considerations

This work studies post-training for mathematical reasoning using public-style benchmark problems, automatically generated model traces, and verifiable answers. It does not involve human subjects, private user data, or sensitive attributes. Because SGSD stores experience-derived skills, care should be taken to keep the skill bank aligned with the

intended training data and to avoid mixing evaluation problems into skill construction. The method also uses verifier outcomes to decide teacher polarity, so applications beyond mathematical reasoning should ensure that the outcome signal is appropriate for the task. As with other reasoning post-training methods, improved mathematical performance should not be interpreted as broad reliability in high-stakes domains without separate validation.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

Xiao Chen, Jiazhen Huang, Zhiming Liu, Qinting Jiang, Fanding Huang, Jingyan Jiang, and Zhi Wang. 2025. Test-Time Distillation for Continual Model Adaptation. *arXiv preprint*.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2026. Does Reinforcement Learning Really Incentivize Reasoning Capacity in Llms Beyond the Base Model? *Advances in Neural Information Processing Systems*, 38:57654–57689.

Fangming Cui, Sunan Li, and Jiahong Li. 2026. A Brief Overview: On-Policy Self-Distillation In Large Language Models. *arXiv preprint*.

Zhengru Fang, Senkang Forest Hu, Zhonghao Chang, Yu Guo, Yihang Tao, Hongyao Liu, Mengzhe Ruan, Jun Huang, and Yuguang Fang. 2026. Inference-Time Budget Control for LLM Search Agents. *arXiv preprint*.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. 2026. Revisiting on-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint*.

Yuhan Guo, Cong Guo, Aiwen Sun, Hongliang He, Xinyu Yang, Yue Lu, Yingji Zhang, Xuntao Guo, Dong Zhang, Jianzhuang Liu, Jiang Duan, Yijia Xiao, Liangjian Wen, Hai-Ming Xu, and Yong Dai. 2026. Web-CogReasoner: Towards Multimodal Knowledge-Induced Cognitive Reasoning for Web Agents. *arXiv preprint*.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. 2026. Self-Distillation Zero: Self-Revision Turns Binary Rewards into Dense Supervision. *arXiv preprint*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-Rank Adaptation of Large Language Models. *arXiv preprint*.

Senkang Hu, Yong Dai, Xudong Han, Zhengru Fang, Yuzhi Zhao, Sam Tak Wu Kwong, and Yuguang Fang. 2026a. Self-Induced Outcome Potential: Turn-Level Credit Assignment for Agents Without Verifiers. *arXiv preprint*.

Senkang Hu, Yong Dai, Yuzhi Zhao, Yihang Tao, Yu Guo, Zhengru Fang, Sam Tak Wu Kwong, and Yuguang Fang. 2026b. Optimizing Agentic Reasoning with Retrieval via Synthetic Semantic Information Gain Reward. *arXiv preprint*.

Senkang Hu, Zhengru Fang, Zihan Fang, Yiqin Deng, Xianhao Chen, and Yuguang Fang. 2024. AgentsCoDriver: Large Language Model Empowered Collaborative Driving with Lifelong Learning. *arXiv preprint*, arXiv:2404.06345.

Senkang Hu, Zhengru Fang, Zihan Fang, Yiqin Deng, Xianhao Chen, Yuguang Fang, and Sam Tak Wu Kwong. 2025. AgentsCoMerge: Large Language Model Empowered Collaborative Decision Making for Ramp Merging. *IEEE Transactions on Mobile Computing*, 24(10):9791–9805.

Senkang Hu, Xudong Han, Jinqi Jiang, Yihang Tao, Zihan Fang, Yong Dai, Sam Kwong, and Yuguang Fang. 2026c. Distribution-Aligned Decoding for Efficient Llm Task Adaptation. *Advances in Neural Information Processing Systems*, 38:56153–56188.

Yixu Huang, Bo Li, Na Li, Zhe Wang, Kaijie Chen, Haonan Ge, Qingyi Si, Yuanzhe Shen, Ruihan Yang, Guangjing Wang, and 1 others. 2026. GUI Agents for Continual Game Generation. *arXiv preprint*.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and 1 others. 2026. Reinforcement Learning via Self-Distillation. *arXiv preprint*.

Yiqiao Jin, Yiyang Wang, Lucheng Fu, Yijia Xiao, Yinyi Luo, Haoxin Liu, B Aditya Prakash, Josiah Hester, Jindong Wang, and Srijan Kumar. 2026. UniSD: Towards a Unified Self-Distillation Framework for Large Language Models. *arXiv preprint*.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. 2026a. Unifying Group-Relative and Self-Distillation Policy Optimization via Sample Routing. *arXiv preprint*.

Quanjiang Li, Zhiming Liu, Wei Luo, Tingjin Luo, and Chenping Hou. 2026b. Correcting Visual Blur Induced by Attention Distraction to Reduce Hallucinations: Algorithm and Theory. *arXiv preprint*.

Weiting Liu, Jieyi Bi, Wanqi Zhou, Jianfeng Feng, Yining Ma, Ai Han, and Wenlian Lu. 2026a. ToolAnchor: Anchoring Counterfactual Context to Boost Agentic Tool-Use Capability. *arXiv preprint*.

Weiting Liu, Han Wu, Yufei Kuang, Xiongwei Han, Tao Zhong, Jianfeng Feng, and Wenlian Lu. 2026b. Automated Optimization Modeling via a Localizable Error-Driven Perspective. *arXiv preprint*.

Zhiming Liu, Yujie Wei, Lei Feng, Xiu Su, Xiaobo Xia, Weili Guan, Zeke Xie, and Shuo Yang. 2026c. Do All Individual Layers Help? An Empirical Study of Task-Interfering Layers in Vision-Language Models. *arXiv preprint*.

Weijian Ma, Ruoxin Chen, Keyue Zhang, Shuang Wu, and Shouhong Ding. 2025. Instruct Where the Model Fails: Generative Data Augmentation via Guided Self-Contrastive Fine-Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5991–5999.

Weijian Ma, Shizhao Sun, Tianyu Yu, Ruiyu Wang, Tat-Seng Chua, and Jiang Bian. 2026. Thinking with Blueprints: Assisting Vision-Language Models in Spatial Reasoning via Structured Object Representation. *arXiv preprint*.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. 2026. Trace2skill: Distill Trajectory-Local Lessons into Transferable Agent Skills. *arXiv preprint*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model Is Secretly a Reward Model. *Advances in neural information processing systems*, 36:53728–53741.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint*.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-Distillation Enables Continual Learning. *arXiv preprint*.

Yibo Shi, Jungang Li, Linghao Zhang, Zihao Dongfang, Biao Wu, Sicheng Tao, Yibo Yan, Chenxi Qin, Weiting Liu, Zhixin Lin, and 1 others. 2026. AndroTMem: From Interaction Trajectories to Anchored Memory in Long-Horizon GUI Agents. *arXiv preprint*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. *Advances in neural information processing systems*, 36:8634–8652.

Xuanbo Su, Wenhao Hu, Haibo Su, Yunzhang Chen, Le Zhan, Yanqi Yang, and Leo Huang. 2026. Sell More, Play Less: Benchmarking LLM Realistic Selling Skill. *arXiv preprint*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint*.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and 1 others. 2026. Skill-Sd: Skill-Conditioned Self-Distillation for Multi-Turn Llm Agents. *arXiv preprint*.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. 2025. Reinforcement Learning for Self-Improving Agent with Skill Library. *arXiv preprint*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, and 1 others. 2026. Skillrl: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning. *arXiv preprint*.

Donglai Xu, Hongzheng Yang, Yuzhi Zhao, Pingping Zhang, Jinpeng Chen, Wenao Ma, Zhijian Hou, Mengyang Wu, Xiaolei Li, Senkang Hu, and 1 others. 2025. From Exploration to Exploitation: A Two-Stage Entropy RLVR Approach for Noise-Tolerant MLLM Training. *arXiv preprint*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 Technical Report. *arXiv preprint*.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. 2026. Self-Distilled RLVR. *arXiv preprint*.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026. On-Policy Context Distillation for Language Models. *arXiv preprint*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An Open-Source Llm Reinforcement Learning System at Scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-Rewarding Language Models. *arXiv preprint*.

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

Zepeng Zhai, Meilin Chen, Jiaxuan Zhao, Junlang Qian, Lei Shen, and Yuan Lu. 2026. Rewards as Labels: Revisiting RLVR from a Classification Perspective. *arXiv preprint*.

Haobo Zhang, Xutao Mao, Guangyuan Dong, Ziwei Li, Xuanbo Su, Kaijie Chen, Jing Yang, and Zheng Lin. 2026a. MemMark: State-Evolution Attribution Watermarking for Agent Long-Term Memory Systems. *arXiv preprint*.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026b. MemSkill: Learning and Evolving Memory Skills for Self-Evolving Agents. *arXiv preprint*.

Linghao Zhang, Jungang Li, Yonghua Hei, Sicheng Tao, Song Dai, Yibo Yan, Zihao Dongfang, Weiting Liu, Chenxi Qin, Hanqian Li, and 1 others. 2026c. Temporal Gains, Spatial Costs: Revisiting Video Fine-Tuning in Multimodal Large Language Models. *arXiv preprint*.

Shuai Zhang, Jiayu Hu, Zijie Chen, Zeyuan Ding, Yi Zhang, Yingji Zhang, Ziyi Zhou, Junwei Liao, Shengjie Zhou, Yong Dai, Zhenzhong Lan, and Xiaozhu Ju. 2026d. A2Eval: Agentic and Automated Evaluation for Embodied Brain. *arXiv preprint*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint*.

A Artifact License

The research artifacts associated with this paper include the training and evaluation code, configuration files, prompt templates, and the generated skill-bank files used by SGSD. These artifacts will be distributed for research use with MIT license in the incoming released repository. Users of the released artifacts should also comply with the licenses and terms of the underlying third-party resources, including the base Qwen3 models, the Qwen3 embedding model, the DAPO-Math-17K training data, the mathematical reasoning benchmarks, and the software libraries used for training and evaluation. The released skill banks are derived from training-time model trajectories and are intended to support reproduction and further research on skill-conditioned self-distillation.

B Use of AI Assistants

AI assistants were used during this project to help with code implementation, debugging, and language polishing. All core research ideas, method design choices, theoretical arguments, experimental designs, result interpretations, and final scientific claims were developed and checked by the authors. The authors reviewed and edited AI-assisted code and text before inclusion, and remain responsible for the correctness, integrity, and presentation of the paper and accompanying artifacts.

C Skill Bank Format

SGSD uses a structured JSON skill bank. As shown in Tab. 5, the implementation stores two knowledge fields: `general_skills` and `common_mistakes`. Each model scale uses a model-specific cold-start bank extracted from DAPO-Math training problems. Fig. 5 gives a concrete serialized example of this schema.

In the released artifacts used by our experiments, the Qwen3-1.7B, Qwen3-4B, and Qwen3-8B banks contain 28/24/46 general skills and 98/15/8 common mistakes, respectively.

D Prompt and Skill Construction Details

This section gives the prompt templates and algorithms used to construct and consume the skill bank. The templates are rendered as plain user messages before the model-specific chat template is applied.

Table 5: JSON fields used by the SGSD skill bank.

Field	Meaning
`general_skills`	Reusable positive guidance. Each entry contains `skill_id`, `title`, `principle`, and `when_to_apply`.
`common_mistakes`	Reusable negative guidance. Each entry contains `mistake_id`, `description`, `why_it_happens`, and `how_to_avoid`.
`metadata`	Provenance and merge statistics, including source description, merge group size, stagnation patience, and layer-level merge counts.

```
{
  "general_skills": [
    {
      "skill_id": "gen_001",
      "title": "Translate Constraints to Algebra",
      "principle": "Convert stated constraints into algebraic relations.",
      "when_to_apply": "When variables are governed by explicit conditions."
    }
  ],
  "common_mistakes": [
    {
      "mistake_id": "err_001",
      "description": "Skipping the constraint model.",
      "why_it_happens": "The solver starts computing before formalizing
          conditions.",
      "how_to_avoid": "Restate the configuration before deriving equations."
    }
  ],
  "metadata": {
    "source": "hierarchical merge from raw candidates",
    "merge_group_size": 32,
    "merge_stagnation_patience": 3
  }
}
```

Figure 5: Example JSON instance for the SGSD skill bank. The line breaks follow the actual serialized structure but use shortened entry text for readability.

D.1 Student and Teacher Contexts

SGSD keeps a strict information boundary. The student rollout and evaluation prompt contain only the problem, while each teacher prompt receives one retrieved skill–mistake guidance block in addition to the problem. Vanilla SGSD setting does not include a reference solution in the teacher prompt. Fig. 6 shows the corresponding student and teacher prompt templates.

D.2 Cold-Start Memory Generation and Skill Extraction

The cold-start pipeline first samples in-domain problems, generates model completions, scores them with the answer verifier, and converts them into compact memories. Successful memories are used to extract general skills, while failed memories are used to extract common mistakes. Alg. 2 summarizes the construction procedure. The three prompt templates used in this stage are shown in Tabs. 6, 7, and 8, respectively.

D.3 Skill Bank Merging

Raw candidate skills are noisy and redundant, so the implementation merges each collection hierarchically. Each merge call receives up to 32 items, preserves unique insights, and returns a compact JSON collection. Merging stops when the root group is reached or when the item count stops decreasing for three consecutive layers. Alg. 3 gives the merge procedure, while Tabs. 9 and 10 give the corresponding merge prompts.

D.4 Online Skill Evolution

The online updater reuses the same success/failure extraction prompts and merge prompts. It updates only the dynamic portion of the bank; static cold-start entries are preserved. Alg. 4 gives the complete online update procedure.

Student Prompt

```
Problem: {problem}

Please reason step by step, and put your final answer within \boxed{}.
```

Teacher Prompt for Pair k

```
You may use the following retrieved math-reasoning guidance as soft guidance.
Solve the current problem independently and do not quote it verbatim.

### General Principles
- **{skill.title}**: {skill.principle}
  _Apply when: {skill.when_to_apply}_

### Mistakes to Avoid
- **Don't**: {mistake.description}
  **Instead**: {mistake.how_to_avoid}

Problem: {problem}

Please reason step by step, and put your final answer within \boxed{}.
```

Figure 6: Student and teacher prompt templates in SGSD. The student receives only the problem, while each teacher scores the same student rollout under one retrieved skill–mistake context.

Prompt: Cold-Start Memory Generation

```
You are a careful mathematical problem-solving agent.
Solve the following problem step by step. Keep the reasoning coherent and self-contained, and
end with a single final answer enclosed in \boxed{{}}.
Problem:
{problem}
```

Table 6: Prompt template for generating cold-start mathematical reasoning memories.

In our implementation, $F = 25$, $\gamma = 0.8$, $N = 5$, and $C = 30$ when dynamic skill updates are enabled. The updater can use either the current training model or a separate generation backend.

E Proof Details

E.1 Derivative of the Gated Loss

The gated token loss is

$$\ell_{\text{gate}}(\Delta) = \log 2 - \log\left(1 + \exp\left(-\frac{\Delta^2}{2\tau_g}\right)\right). \tag{27}$$

Let $z(\Delta) = -\Delta^2/(2\tau_g)$, then

$$\frac{\partial \ell_{\text{gate}}(\Delta)}{\partial \Delta} = -\frac{\exp(z(\Delta))}{1+\exp(z(\Delta))}\frac{\partial z(\Delta)}{\partial \Delta} \tag{28}$$

$$= \frac{\Delta}{\tau_g}\frac{\exp(-\Delta^2/(2\tau_g))}{1+\exp(-\Delta^2/(2\tau_g))} \tag{29}$$

$$= \frac{\Delta}{\tau_g\left(1+\exp(\Delta^2/(2\tau_g))\right)}. \tag{30}$$

This gives the gradient factor

$$g_\tau(\Delta) = \frac{\Delta}{\tau_g\left(1+\exp(\Delta^2/(2\tau_g))\right)}. \tag{31}$$

E.2 Shape of the Gated Objective

As shown in Fig. 7, the gated loss $\ell_{\text{gate}}(\Delta)$ is symmetric and bounded: negligible teacher-student gaps receive little weight, while very large gaps

Prompt: General-Skill Extraction from a Successful Memory

```
You are an expert at distilling mathematical reasoning behavior into concise, reusable skills
for a reinforcement-learning agent.
You will be given ONE successful math problem-solving memory. The memory contains the original
problem, a compact reasoning trajectory, and a summarized raw attempt.
Your task:
1. Derive 1-3 GENERAL skills that likely contributed to the success.
2. Each skill must be broadly reusable across algebra, geometry, number theory, combinatorics,
and olympiad-style reasoning.
3. Phrase each skill as an actionable principle; avoid task-specific constants, entity names, or
one-off details unless they express a general method.
4. Merge overlapping ideas inside this response; do not output near-duplicate skills.
5. Use only evidence grounded in the provided memory.
Successful memory:
{memory_json}
Return ONLY valid JSON with key general_skills.
```

Table 7: Prompt template for extracting reusable positive skills from successful cold-start memories.

Prompt: Common-Mistake Extraction from a Failed Memory

```
You are an expert at analyzing failed mathematical reasoning and turning failures into concise,
reusable cautionary skills for a reinforcement-learning agent.
You will be given ONE failed math problem-solving memory. The memory contains the original
problem, summarized failure evidence, and the raw final attempt.
Your task:
1. Derive 1-3 COMMON mistakes that explain the failure.
2. Each item must describe a general failure mode, why it happens, and how to avoid it in future
math reasoning.
3. Make every item broadly reusable across algebra, geometry, number theory, combinatorics, and
olympiad-style reasoning.
4. Merge overlapping ideas inside this response; do not output near-duplicate mistakes.
5. Use only evidence grounded in the provided memory.
Failed memory:
{memory_json}
Return ONLY valid JSON with key common_mistakes.
```

Table 8: Prompt template for extracting reusable mistake patterns from failed cold-start memories.

saturate instead of growing without limit. Its derivative $g_\tau(\Delta)$ preserves the sign of the log-probability gap Δ in the informative region, but decays toward zero for extreme gaps. Thus, the gate keeps medium teacher-student disagreements as useful dense supervision while preventing outlier logits or prompt artifacts from dominating the update.

E.3 Policy-Gradient Form

For teacher k and token t, recall that $\Delta_t^{(k)} = \log p_T^{(k)}(y_t \mid x, y_{<t}) - \log p_S(y_t \mid x, y_{<t})$. Since teacher logits are stop-gradient targets,

$$\nabla_\theta \Delta_t^{(k)} = -\nabla_\theta \log p_S(y_t \mid x, y_{<t}). \tag{32}$$

Let $Z = \sum_t m_t + \epsilon$. For $\bar{\ell}^{(k)} = \frac{1}{Z}\sum_{t=1}^{T} m_t \ell_{\text{gate}}(\Delta_t^{(k)})$, we have

$$\nabla_\theta \bar{\ell}^{(k)} = -\frac{1}{Z}\sum_{t=1}^{T} m_t g_\tau(\Delta_t^{(k)}) \cdot \nabla_\theta \log p_S(y_t \mid x, y_{<t}). \tag{33}$$

Substituting this into Eq. (16) yields

$$-\nabla_\theta \mathcal{L}_{\text{SGSD}}(x) = \sum_{k=1}^{K_x}\sum_{t=1}^{T} \alpha_k(x)\rho_k \frac{m_t}{Z} g_\tau(\Delta_t^{(k)}) \cdot \nabla_\theta \log p_S(y_t \mid x, y_{<t}). \tag{34}$$

Thus, SGSD is a policy-gradient-style update whose dense credit is controlled by teacher polarity and the gated gap derivative.

Prompt: General-Skill Merging

```
You are an expert at consolidating independently-generated math skills into a compact,
non-redundant skill bank.
You will be given up to 32 general skills extracted from different memories. Some are duplicates,
some partially overlap, and some are unique.
Your task:
1. Merge semantically duplicate or strongly overlapping skills.
2. Preserve all unique insights.
3. Prefer the most general, transferable wording.
4. Treat recurrence as evidence that the pattern is systematic and synthesize one stronger skill.
5. Do not force a fixed final count.
6. Do not mention specific problems, source memories, or dataset names.
General skills to merge:
{items_json}
Return ONLY valid JSON with key general_skills.
```

Table 9: Prompt template for hierarchically merging general-skill candidates.

Prompt: Common-Mistake Merging

```
You are an expert at consolidating independently-generated math failure lessons into a compact,
non-redundant caution bank.
You will be given up to 32 common-mistake items extracted from different memories. Some are
duplicates, some partially overlap, and some are unique.
Your task:
1. Merge semantically duplicate or strongly overlapping mistakes.
2. Preserve all unique insights.
3. Prefer the most general, transferable wording.
4. Treat recurrence as evidence that the failure pattern is systematic and synthesize one stronger
mistake item.
5. Do not force a fixed final count.
6. Do not mention specific problems, source memories, or dataset names.
Common mistakes to merge:
{items_json}
Return ONLY valid JSON with key common_mistakes.
```

Table 10: Prompt template for hierarchically merging common-mistake candidates.

E.4 Sign and Bounded Influence

For $\Delta \neq 0$, the denominator of $g_\tau(\Delta)$ is positive, so

$$\operatorname{sign}(g_\tau(\Delta)) = \operatorname{sign}(\Delta). \tag{35}$$

The outcome-derived polarity ρ_k therefore decides whether a teacher's token-level support is distilled or reversed. For boundedness,

$$1 + \exp\left(\frac{\Delta^2}{2\tau_g}\right) \geq \exp\left(\frac{\Delta^2}{2\tau_g}\right), \tag{36}$$

and hence

$$|g_\tau(\Delta)| \leq \frac{|\Delta|}{\tau_g} \exp\left(-\frac{\Delta^2}{2\tau_g}\right). \tag{37}$$

The right-hand side is maximized at $|\Delta| = \sqrt{\tau_g}$, which gives

$$|g_\tau(\Delta)| \leq \frac{1}{\sqrt{e\,\tau_g}}. \tag{38}$$

Therefore, the token-level coefficient induced by the gate cannot grow without bound as teacher-student gaps become extreme.

E.5 Local Approximation

Near zero, the expansion of g_τ is

$$g_\tau(\Delta) = \frac{\Delta}{2\tau_g} + O\left(\frac{\Delta^3}{\tau_g^2}\right), \tag{39}$$

and the loss itself satisfies

$$\ell_{\text{gate}}(\Delta) = \frac{\Delta^2}{4\tau_g} + O\left(\frac{\Delta^4}{\tau_g^2}\right). \tag{40}$$

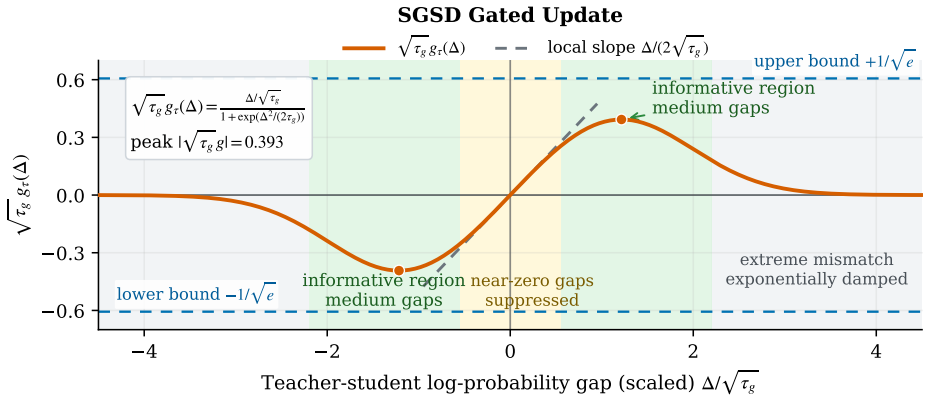

Figure 7: **Shape of the gated objective.**

Algorithm 2 Cold-start skill construction

Require: Seed set $\mathcal{D}_0$, base model π_{θ_0}, verifier, extraction prompts
Initialize memory set $\mathcal{M} \leftarrow \emptyset$
for each problem $x \in \mathcal{D}_0$ **do**
  Render the memory-generation prompt and sample a completion from π_{θ_0}
  Extract the boxed answer and compute the verifier reward
  Build a memory record with problem, completion, reward, answer, trajectory summary, and feedback
  Add the memory record to $\mathcal{M}$
end for
Split $\mathcal{M}$ into successful memories $\mathcal{M}^+$ and failed memories $\mathcal{M}^-$
for each memory $m \in \mathcal{M}^+$ **do**
  Extract up to three general-skill candidates with fields `title`, `principle`, and `when_to_apply`
end for
for each memory $m \in \mathcal{M}^-$ **do**
  Extract up to three common-mistake candidates with fields `description`, `why_it_happens`, and `how_to_avoid`
end for
Merge and re-index the two candidate collections using Alg. 3
Return the initialized skill bank $\mathcal{B} = \{\mathcal{G}, \mathcal{E}\}$

Thus negligible teacher-student gaps induce only small updates. For large gaps,

$$g_\tau(\Delta) = \frac{\Delta}{\tau_g} \exp\left(-\frac{\Delta^2}{2\tau_g}\right)(1+o(1)), \quad |\Delta| \to \infty, \tag{41}$$

so extreme mismatches are exponentially damped.

Finally, consider locally close teacher and student distributions at a fixed history h_t, and define

$$\Delta_v = \log p_T(v \mid h_t) - \log p_S(v \mid h_t). \tag{42}$$

After the first-order normalization term cancels,

Algorithm 3 Hierarchical skill merging

Require: Candidate collection $\mathcal{C}$, prompt type q, group size $H = 32$, stagnation patience $P = 3$
Set layer index $\ell \leftarrow 0$ and stagnant counter $s \leftarrow 0$
repeat
  Partition $\mathcal{C}$ into groups of at most H items
  for each group with more than one item **do**
    Render the merge prompt for type q and parse the returned JSON
    Fall back to the original group if parsing fails
  end for
  Concatenate all merged groups into $\mathcal{C}'$
  Update $s \leftarrow 0$ if $|\mathcal{C}'| < |\mathcal{C}|$, otherwise $s \leftarrow s + 1$
  Set $\mathcal{C} \leftarrow \mathcal{C}'$ and $\ell \leftarrow \ell + 1$
until there is one group or $s \geq P$
Remove exact duplicates and reassign IDs with prefix `gen` or `err`
Return the merged collection

reverse KL has the second-order expansion

$$D_{\mathrm{KL}}(p_T \,\|\, p_S) = \frac{1}{2}\sum_v p_T(v \mid h_t)\Delta_v^2 + O(\|\Delta\|^3). \tag{43}$$

Its local gradient is therefore linear in the log-probability gap. The gated objective matches this local linear behavior up to the constant scale $1/(2\tau_g)$, while deliberately departing from reverse KL for extreme gaps.

F Experimental Details

F.1 Shared and Baseline Configurations

The main experiments use Qwen3-1.7B, Qwen3-4B, and Qwen3-8B with LoRA adaptation. Training uses the English subset of DAPO-Math-17K, and evaluation is performed on AIME24, AIME25, and HMMT25. All trainable methods use learning rate 5×10^{-6}, LoRA rank 64, LoRA alpha 128, bfloat16 training, FlashAttention 2, and gradient checkpointing. Evaluation uses 12 samples

Algorithm 4 Online skill evolution (update and maintenance)

Require: Recent rollout records $\mathcal{W}$, current bank $\mathcal{B}$, update frequency F, success threshold γ
Require: Max new items N, dynamic capacity C
if online update is disabled, $\mathcal{W}$ is empty, or the current step is not divisible by F **then**
    Return $\mathcal{B}$
end if
Gather rollout records across workers and compute success rate $\hat{p}$
if $\hat{p} \geq \gamma$ **then**
    Return $\mathcal{B}$
end if
Split $\mathcal{W}$ into successful records $\mathcal{W}^+$ and failed records $\mathcal{W}^-$
Extract new general-skill candidates from $\mathcal{W}^+$
Extract new common-mistake candidates from $\mathcal{W}^-$
Merge new candidates of each type
Merge the new dynamic candidates with existing dynamic entries
Keep at most N net new dynamic entries per update and at most C dynamic entries per collection
Replace the dynamic subset in $\mathcal{B}$ and save the latest bank and a step-indexed snapshot

per problem, temperature 1.0, top-p 0.95, no top-k truncation, max new tokens 38912, and symbolic answer checking with a string-based fallback. Tab. 11 lists the method-specific hyperparameters that differ across baselines.

F.2 SGSD-Specific Configuration

Tab. 12 lists the SGSD-specific settings used by the final trainer. Named ablations change only the corresponding component, such as disabling token masking, removing polarity, using a single teacher, replacing the gated loss, or switching the teacher update strategy.

Table 11: Key training hyperparameters by method. Empty entries indicate that the parameter is not used by that method.

Hyperparameter	OPSD	OPSD+Skill	GRPO	GRPO+Skill	SGSD
Completion length	1024	1024	16000	16000	1024
Prompt length	20000	20000	2048	2048	20000
Temperature	1.1	1.1	1.2	1.2	1.1
Top-p / top-k	0.95 / 20	0.95 / 20	–	–	0.95 / 20
Generations per prompt	1	1	8	8	1
Teacher count per prompt	1	1	–	–	8
Teacher update	fixed	fixed	–	–	live (synchronized)
Reference solution	yes	yes	–	–	no

Table 12: SGSD-specific hyperparameters.

Component	Setting
Cold-start bank	256 in-domain training problems per model scale
Retrieval encoder	Qwen3-Embedding-0.6B
Retrieved guidance	top-8 general skills and top-8 common mistakes
Teacher pool	one rank-aligned skill–mistake pair per teacher
Teacher weighting	softmax-normalized retrieval scores
Teacher policy	live synchronized self-teacher
Privileged information	skill–mistake guidance only; no reference solution
Vocabulary support	full-vocabulary normalization
Gated objective	$\tau_g = 1.0$
Robust support estimation	token masking, gap clipping with $c_\Delta = 3.0$, confidence threshold $\epsilon_a = 0.05$
Online skill update	$F = 25$, $\gamma = 0.8$, $N = 5$, $C = 30$