CLOSING THE SELF-TEACHING LOOP FOR OPEN-ENDED GENERATION: SIBLING-CONDITIONED DISTILLATION MEETS SELF-VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-teaching improves language models by learning from their own generations and assessments, but extending this loop to open-ended generation remains challenging because response quality is multidimensional and task-native verification is generally unavailable. Rubrics provide a natural specification in this setting, but do not by themselves close the learning loop: how should a fixed rubric be grounded in the model's current responses, and how should stronger completed responses be distinguished from weaker ones? We introduce **Sibling and Self-Verification Policy Optimization (S^2PO)**, which combines local and global routes to address these challenges. The local route uses same-prompt siblings to ground rubric guidance in current-policy realizations, while the global route reuses rubric-conditioned forward pass to derive continuous response scores for group-relative optimization. Across Qwen3-4B and Qwen3-8B on medical and science tasks, S^2PO improves over the base model by 3.09–5.26 points and outperforms both RGSD and 32B-judge GRPO in all four in-domain settings. These gains persist under out-of-distribution evaluation, with improvements of 2.46–5.14 points over the base model across both model scales.

1 INTRODUCTION

Language models increasingly generate training experience and learning signals for their own post-training (Chen et al., 2026). One direct regime is training-time same-model self-teaching, in which no external model supplies learned supervision. On-policy self-distillation (OPSD) realizes its local component: an acting policy samples trajectories, while a privileged teacher derived from the same initialization revisits policy-sampled prefixes to supply dense next-token targets (Zhao et al., 2026; Song & Zheng, 2026). Trajectory-level feedback operates at a different granularity, orienting updates across completed attempts (Yang et al., 2026). In verifiable tasks like mathematics or code, these functions close the loop naturally: successful attempts or their feedback can condition the teacher, while exact-answer checks or executable tests determine which attempts to reinforce (Zhao et al., 2026; Hübotter et al., 2026; Yang et al., 2026). Open-ended generation lacks such task-native closure: multiple responses may be valid, quality is often multidimensional and prompt dependent, and no exact-match criterion or executable test generally selects among completed responses (Bablani et al., 2026; Gunjal et al., 2025). Extending same-model self-teaching to these tasks therefore requires closing the loop without relying on task-native verification.

For local teaching, rubrics provide privileged context: they describe the qualities of a strong response without prescribing a single answer, and recent OPSD methods condition teachers on them (Rezaei et al., 2026; Gu et al., 2026; Bablani et al., 2026). Yet rubric-only privilege can leave a ***specification–realization gap***: knowing what a response should satisfy does not by itself determine how those requirements should translate into local supervision for the response at hand. In self-distillation, this distinction matters because the learner is updated through the teacher signal induced by the rubric, rather than by the rubric itself. Effective local teaching therefore requires grounding a fixed criterion in concrete guidance for the response being taught.

Even well-grounded local guidance leaves a second ***likelihood–direction gap***: privileged next-token shifts reveal how conditioning changes local continuations, but not whether a completed response

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.

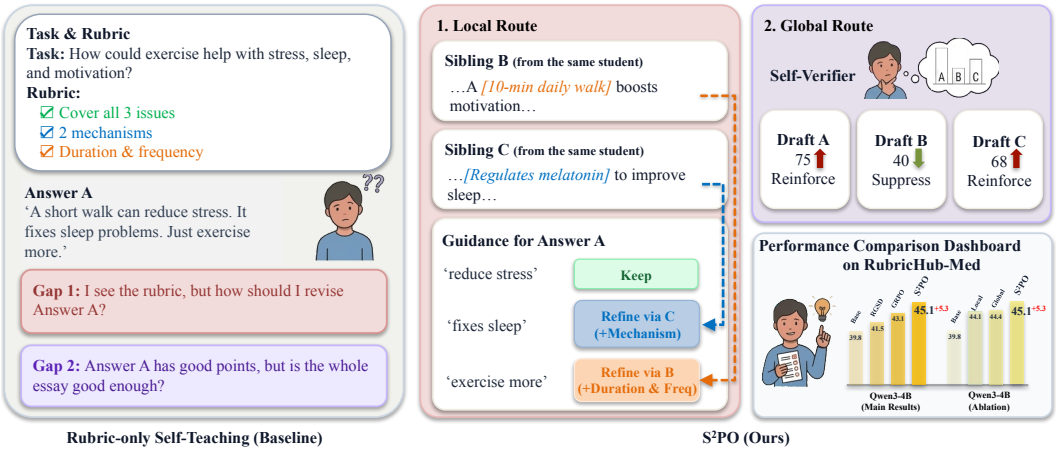


Figure 1: **Two gaps, two routes.** A rubric states what a response should achieve without determining what to change in a particular response (Gap 1), as illustrated by Answer A's missing rubric elements; meanwhile, dense likelihood shifts do not indicate which completed responses should be reinforced (Gap 2). S^{2}PO reuses one same-prompt response group: siblings provide concrete context for *local* revision guidance, e.g., B contributes duration and frequency while C adds a mechanism; self-verification then scores the completed drafts and supplies *global* trajectory direction.

should be reinforced (Nguyen et al., 2026b). Since every rollout can yield dense targets, useful local supervision may coexist with an incomplete or otherwise poor response. Learning must therefore decide not only how to revise visited prefixes, but which completed attempts should shape future policy behavior. Without response-level outcomes, dense teaching alone cannot make this choice. Verifiable tasks obtain this direction from automated validation(Hübotter et al., 2026; Yang et al., 2026), whereas open-ended generation lacks a native counterpart in self-teaching loops.

To address these gaps, we introduce **Sibling and Self-Verification Policy Optimization (S^{2}PO)**, a framework combining sibling-conditioned distillation with self-verification, which reuses each on-policy response group through two complementary post-hoc routes. **Local route.** S^{2}PO pairs a fixed normative rubric with *score-independent descriptive siblings*—alternative responses sampled by the current policy without requiring external quality labels. The rubric specifies desired response properties, while siblings expose additional same-prompt realizations that contextualize its guidance. Repurposing the within-prompt group used by Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as behavioral context, S^{2}PO uses source-matched views to isolate how sibling context refines rubric guidance, fusing corrections computed with and without siblings into a single teacher target for the prompt-only policy. **Global route.** S^{2}PO introduces a *pass-reusing logit self-verifier*: by appending a compact rubric query to the end of the teacher sequence, the same forward pass produces a probability distribution over score tokens, yielding a continuous quality score via expected utility. These scores are normalized within each prompt group to obtain GRPO advantages, providing trajectory guidance for whole responses. The two routes remain score-isolated: verifier scores do not enter dense construction, and sibling text does not enter scoring. A single model frozen at policy initialization supplies both post-hoc interfaces; only the original-prompt policy is updated.

Our contributions are threefold:

1. We formulate two connected gaps in open-ended same-model self-teaching: a specification–realization gap between what a rubric requires and what should change in a given response, and a likelihood–direction gap between privileged next-token shifts and completed-response update direction.

2. We introduce S^{2}PO, whose sibling-conditioned local route repurposes same-prompt response groups as behavioral context through source-matched rubric corrections, while its pass-reusing logit self-verifier maps ordered score-token distributions to continuous global signals for group-relative policy updates, with two routes isolated from each other.

3. S^{2}PO achieves state-of-the-art self-teaching performance across in-domain and transfer benchmarks, backed by controlled factorial and diagnostic analyses of both routes.

2 Related Work

2.1 On-Policy Self-Distillation for Open-ended Generation

On-policy self-distillation (OPSD) uses the model under privileged context as a self-teacher, converting auxiliary information into dense supervision on policy-generated trajectories (Zhao et al., 2026). In open-ended generation, rubrics provide a normative specification of response quality, while recent work further incorporates policy-generated responses as behavioral context for this supervision (Rezaei et al., 2026; Gu et al., 2026; Bablani et al., 2026; Yu et al., 2026; Wu et al., 2026; Li et al., 2026a; Guo et al., 2026). As these conditioning views become richer, however, induced teacher shifts risk altering behavior beyond the intended rubric correction. In parallel, related work explores dynamically adapting criterion weights or the rubric itself as the policy evolves (Wang et al., 2026; Nguyen et al., 2026a; Ichihara et al., 2026; Pan et al., 2026; Tyagi et al., 2026; Guan et al., 2026). Together, these developments move the design question from whether privileged guidance is available to how normative specification and current-policy behavior should interact in local supervision. S^{2}PO makes this interface explicit through a separation of roles. The rubric provides fixed normative criteria, while same-prompt siblings provide descriptive context of current policy behavior. By contrasting matched views with and without siblings, S^{2}PO grounds rubric guidance in real-world policy attempts before distilling it into the prompt-only model, so local supervision reflects both task specification and current-policy realization.

2.2 Rubric-Based Reinforcement Learning and Self-Evaluation

Rubric-based RL provides trajectory-level supervision for open-ended generation by turning multidimensional criteria into response-level rewards (Gunjal et al., 2025). Hybrid methods retain trajectory-level optimization as the global backbone while using token-level supervision as localized refinement (Yang et al., 2026; Xia et al., 2026; Lv et al., 2026). In a self-teaching loop, this naturally raises the question of how the global orienting signal itself is obtained. Existing self-rewarding and verifier-based methods obtain such signals from generated judgments or score-token distributions (Yuan et al., 2024; Kwok et al., 2026). S^{2}PO obtains this signal through self-verification, reusing the rubric-conditioned computation from its local route: a lightweight scoring readout over the frozen pass provides trajectory-level guidance without additional generative judging.

3 Preliminaries

Rubric-based reinforcement learning. Let $x \sim \mathcal{D}$ be an open-ended prompt with a fixed offline rubric $\mathcal{C}(x) = \{(c_i, \omega_i)\}_{i=1}^{K}$, where c_i is criterion text, $\omega_i \geq 0$, and $\sum_i \omega_i > 0$. We denote its criterion-text view by $\mathcal{C}_{\text{text}}(x) = (c_1, \ldots, c_K)$. Given a completed response y, a rubric evaluator assigns criterion-wise scores $r_i(x, y)$, aggregated as (Gunjal et al., 2025; Xia et al., 2026)

$$s(x, y) = \frac{\sum_{i=1}^{K} \omega_i r_i(x, y)}{\sum_{i=1}^{K} \omega_i}.$$

We instantiate rubric-based reinforcement learning with Group Relative Policy Optimization (GRPO; Shao et al., 2024). At each update, set $\theta_{\text{old}} \leftarrow \theta$ and let $\pi_{\theta_{\text{old}}}$ be the rollout policy. For each prompt, GRPO samples an on-policy response group

$$\mathcal{Y}(x) = \{y_g\}_{g=1}^{G}, \qquad y_g \overset{\text{i.i.d.}}{\sim} \pi_{\theta_{\text{old}}}(\cdot \mid x). \tag{1}$$

Writing $s_g = s(x, y_g)$, and letting $\bar{s}$ and $\widehat{\sigma}_s$ denote the mean and standard deviation of $\{s_g\}_{g=1}^{G}$, GRPO forms the group-relative advantage

$$\widehat{A}_g = \frac{s_g - \bar{s}}{\widehat{\sigma}_s + \epsilon_A}, \qquad \epsilon_A > 0. \tag{2}$$

The policy is then optimized relative to $\pi_{\theta_{\text{old}}}$ using the standard clipped importance-ratio objective. Section 4.2 specifies how S^{2}PO constructs the response scores s_g used in this update.

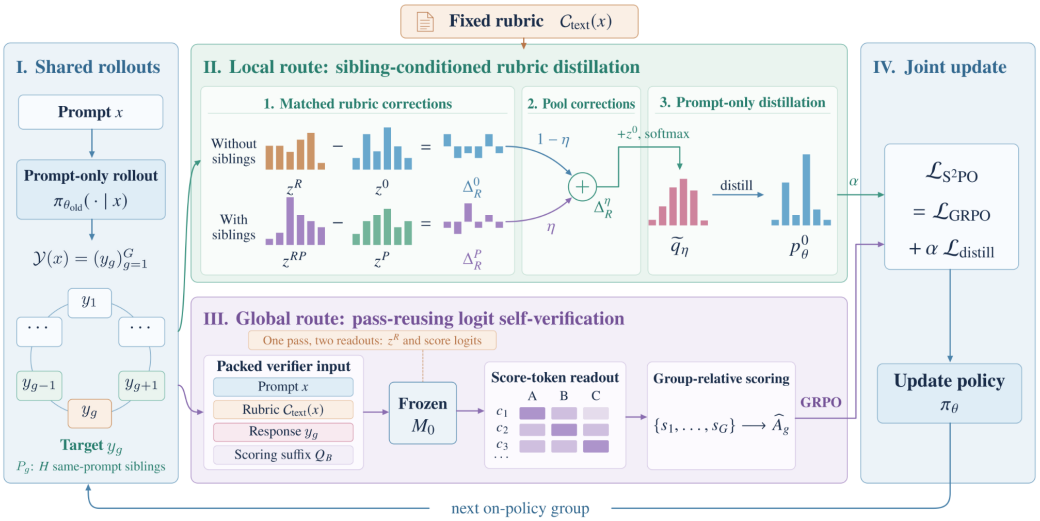


Figure 2: **Overview of S^{2}PO.** (I) Responses are sampled from the original prompt. (II) The local route compares matched rubric corrections with and without sibling context to construct dense targets. (III) The global route extends the rubric-conditioned input with a scoring suffix, reusing one frozen pass for response-position and criterion-score readouts. (IV) Distillation and group-relative reinforcement jointly update the same prompt-only policy.

On-policy self-distillation. Given $y \sim \pi_\theta(\cdot \mid x)$, OPSD revisits prefixes $y_{<t}$ from the current policy using a same-model teacher with privileged context ξ (Zhao et al., 2026). At token t, let

$$p_{\theta,t} = \pi_\theta(\cdot \mid x, y_{<t}), \qquad q_t^\xi = T(\cdot \mid x, \xi, y_{<t})$$

denote prompt-only student and privilege-conditioned teacher distributions. An OPSD objective is

$$\ell_{\text{OPSD}}(y;\xi) = \frac{1}{|y|}\sum_{t=1}^{|y|} D\Big(\text{sg}(q_t^\xi)\,\Big\|\,p_{\theta,t}\Big),$$

where D is a token-level divergence and sg denotes stop-gradient. Thus, privileged information shapes training targets without changing the policy's generation-time conditioning.

4 Method

As shown in Figure 2, we introduce **Sibling and Self-Verification Policy Optimization (S^{2}PO)**, which combines sibling-conditioned rubric distillation with trajectory-level self-verification. The local route uses same-prompt siblings as alternative realizations of the current policy to contextualize rubric guidance and construct dense supervision for the prompt-only policy (Section 4.1). The global route reuses rubric-conditioned computation to convert criterion-level evidence into continuous response scores for group-relative trajectory optimization (Section 4.2).

4.1 Sibling-Conditioned Rubric Distillation

For the ***specification–realization gap***, we compare rubric-induced corrections with and without same-prompt sibling context; their difference isolates the realization-dependent component of the frozen teacher's guidance. Pooling these corrections yields a realization-conditioned target for forward-KL distillation.

Sibling context. For each target y_g in the shared on-policy group $\mathcal{Y}(x)$, we select H other responses from that group as its siblings, with $1 \le H \le G-1$ (default $H = 2$), and serialize them as P_g. Notably, assignment uses neither verifier scores nor quality judgments; siblings provide behavioral context, not quality labels or imitation targets.

Two matched rubric corrections. Let M_0 be a copy of the initial policy checkpoint, frozen throughout training. At each target prefix $(x, y_{g,<t})$, we compare its full-vocabulary logits before and after adding the same rubric. Without siblings, these are $z^0_{g,t}$ and $z^R_{g,t}$; with P_g held fixed, they are $z^P_{g,t}$ and $z^{RP}_{g,t}$. All four views share the target prefix; within each pair, adding the rubric is the only change. Writing $q^V_{g,t} = \mathrm{softmax}(z^V_{g,t})$ for each view, the two matched rubric corrections are

$$\Delta^0_{R,g,t} = z^R_{g,t} - z^0_{g,t}, \qquad \Delta^P_{R,g,t} = z^{RP}_{g,t} - z^P_{g,t}. \tag{3}$$

These are the rubric-only correction and its sibling-conditioned counterpart. Their difference

$$\Delta^{\mathrm{real}}_{R,g,t} = \Delta^P_{R,g,t} - \Delta^0_{R,g,t} \tag{4}$$

captures how current-policy sibling realizations change the rubric's local guidance. It supplies a realization-dependent adjustment to the induced correction while the rubric specification itself remains fixed. Serialization and alignment details appear in Appendix C.1.

Pooled rubric target. To guide the prompt-only learner, we transfer rubric corrections to the frozen prompt-only baseline $z^0_{g,t}$. Directly distilling from $q^{RP}_{g,t}$ would conflate both the sibling-conditioned rubric correction and the sibling-only shift $z^P_{g,t} - z^0_{g,t}$, biasing the student toward sibling context it cannot access at test time. We instead adjust the rubric-only correction with strength η:

$$\begin{aligned} \Delta^\eta_{R,g,t} &= \Delta^0_{R,g,t} + \eta\, \Delta^{\mathrm{real}}_{R,g,t} = (1-\eta)\Delta^0_{R,g,t} + \eta \Delta^P_{R,g,t}, \qquad 0 \le \eta \le 1, \\ \widetilde{q}_{\eta,g,t} &= \mathrm{softmax}\Big(z^0_{g,t} + \Delta^\eta_{R,g,t}\Big). \end{aligned} \tag{5}$$

Thus η scales the realization-dependent adjustment in logit space, equivalently interpolating between the two matched rubric corrections before normalization. At $\eta = 0$, the target is q^R; at $\eta = 1$, the full sibling-conditioned correction $\Delta^P_{R,g,t}$ is transferred to $z^0_{g,t}$. Appendix D.1 gives the target properties.

Prompt-only distillation. The live student distribution is $p_{\theta,g,t} = \pi_\theta(\cdot \mid x, y_{g,<t})$. Let $\mathcal{M}_g$ be the nonempty set of valid target positions in y_g. Writing sg for stop-gradient, we minimize one full-vocabulary forward-KL objective:

$$\mathcal{L}_{\mathrm{distillation}} = \mathbb{E}_{x,\mathcal{Y}}\left[\frac{1}{G}\sum_{g=1}^{G}\frac{1}{|\mathcal{M}_g|}\sum_{t\in\mathcal{M}_g} D_{\mathrm{KL}}(\mathrm{sg}[\widetilde{q}_{\eta,g,t}] \| p_{\theta,g,t})\right]. \tag{6}$$

Rubric and sibling context enter only post-hoc target construction; gradients update the live policy with the prompt-only conditioning used for rollout and deployment.

4.2 Pass-Reusing Trajectory Self-Verification

The ***likelihood–direction gap*** remains after local distillation: next-token guidance alone does not specify which completed responses should be reinforced. We address it by evaluating each rollout against the rubric and using the resulting continuous scores for group-relative reinforcement.

Rubric-conditioned pass reuse. To obtain this evaluation, we extend the rubric-conditioned input already used by the frozen model M_0 to compute z^R. After each completed response y_g, we append a fixed suffix Q_B containing a verifier query and one score slot per rubric criterion. Causal masking preserves the earlier response-position logits z^R.

We prefill every score slot with the midpoint label `B` (under the utilities defined below), establishing a deterministic scorecard history rather than autoregressively generating earlier verdicts. The extended frozen pass therefore supplies two readout regions: response positions provide z^R, and positions immediately before the score tokens provide criterion-level scoring logits (Figure 2). Query, serialization, and score-slot details appear in Appendix C.2.

Distributional criterion scoring. We score each criterion from its label distribution rather than a hard verdict. Let $\mathcal{V}_{\mathrm{score}} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$ denote satisfied, partially satisfied or uncertain, and not satisfied, respectively, with utilities $\phi(\mathtt{A}) = 1$, $\phi(\mathtt{B}) = \frac{1}{2}$, and $\phi(\mathtt{C}) = 0$. For label ℓ, let $z_{g,i,\ell}$ be its

next-token logit immediately before criterion i's fixed score token. Restricting and renormalizing over the three labels gives

$$\nu_{g,i,\ell} = \frac{\exp(z_{g,i,\ell})}{\sum_{\ell' \in \mathcal{V}_{\text{score}}} \exp(z_{g,i,\ell'})}, \qquad \ell \in \mathcal{V}_{\text{score}}. \tag{7}$$

The prefilled B tokens fix the scorecard history, whereas $\nu_{g,i,\mathtt{B}}$ is the probability predicted at criterion i. We take the expected utility as the criterion score and aggregate it with the rubric weights:

$$r_i(x, y_g) = \sum_{\ell \in \mathcal{V}_{\text{score}}} \nu_{g,i,\ell}\, \phi(\ell), \qquad s_g = s(x, y_g) = \frac{\sum_{i=1}^{K} \omega_i r_i(x, y_g)}{\sum_{i=1}^{K} \omega_i}. \tag{8}$$

Criterion readouts with the same argmax label can therefore receive different continuous scores.

Group-relative trajectory update. Applying the GRPO normalization in Eq. 2 to $\{s_g\}_{g=1}^{G}$ gives advantages $\widehat{A}_g$ that are positive for above-mean responses and negative for below-mean ones. Each response's advantage is shared across its valid token positions. With these symmetric utilities, group centering cancels the common $1/2$ midpoint, leaving trajectory direction governed by the balance between positive (A) and negative (C) evidence (More details in Appendix Proposition D.1).

For $t \in \mathcal{M}_g$, define the original-prompt importance ratio $\rho_{g,t} = \frac{\pi_\theta(y_{g,t}|x, y_{g,<t})}{\pi_{\theta_{\text{old}}}(y_{g,t}|x, y_{g,<t})}$. With clipping radius $\epsilon > 0$, we minimize

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{\substack{x \sim \mathcal{D} \\ y_{1:G} \sim \pi_{\theta_{\text{old}}}(\cdot|x)}} \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{|\mathcal{M}_g|} \sum_{t \in \mathcal{M}_g} \min\Big(\rho_{g,t} \widehat{A}_g,\ \text{clip}(\rho_{g,t}, 1-\epsilon, 1+\epsilon) \widehat{A}_g \Big) \right]. \tag{9}$$

Algorithm 1 Training procedure for S^{2}PO.

Require: $(\mathcal{D}, \mathcal{C})$, policy π_θ, frozen M_0, sizes G, H, weights η, α

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$; sample $x \sim \mathcal{D}$; $R \leftarrow \mathcal{C}_{\text{text}}(x)$
  $\mathcal{Y} = (y_g)_{g=1}^{G} \leftarrow \textsc{Rollout}(\pi_{\theta_{\text{old}}}, x, G)$
  $P_{1:G} \leftarrow \textsc{AssignSiblings}(\mathcal{Y}, H)$
  for $g = 1, \ldots, G$ **do**
    $z_g^0, z_g^P \leftarrow M_0(x, y_g),\ M_0(x, P_g, y_g)$
    $(z_g^R, v_g), z_g^{RP} \leftarrow M_0(x, R, y_g, Q_B),\ M_0(x, R, P_g, y_g)$
    $\widetilde{q}_g \leftarrow \textsc{PooledRubricTarget}(z_g^0, z_g^P, z_g^R, z_g^{RP}; \eta)$
    $s_g \leftarrow \textsc{SelfVerify}(v_g, \omega_{1:K})$
  end for
  $\widehat{A}_{1:G} \leftarrow \textsc{GroupNormalize}(s_{1:G})$
  $\mathcal{L}_{\text{distillation}} \leftarrow \textsc{Distill}(\pi_\theta, x, \mathcal{Y}, \widetilde{q}_{1:G})$
  $\mathcal{L}_{\text{GRPO}} \leftarrow \textsc{GRPO}(\pi_\theta, \pi_{\theta_{\text{old}}}, x, \mathcal{Y}, \widehat{A}_{1:G})$
  $\theta \leftarrow \textsc{Update}(\theta, \mathcal{L}_{\text{GRPO}} + \alpha \mathcal{L}_{\text{distillation}})$
end for

$v_g = (z_{g,i,\ell})_{i,\ell}$ collects score-slot logits; z_g^V and $\widetilde{q}_g$ collect response-position logits and targets.

4.3 Joint Training

The dense and trajectory objectives share the rollout distribution in Eq. 1 and combine as

$$\mathcal{L}_{\text{S}^2\text{PO}} = \mathcal{L}_{\text{GRPO}} + \alpha \mathcal{L}_{\text{distillation}}, \qquad \alpha \geq 0. \tag{10}$$

The coefficient α balances dense distillation against the trajectory update; η instead controls the sibling-induced adjustment within the dense target (Section 4.1). We accumulate both policy gradients before one optimizer step updates θ.

The two routes share the rollout group, rubric, frozen model M_0, and live policy, while their training signals are constructed separately. Sibling assignment and dense-target construction never consult verifier scores, and verification never reads sibling text. These routes only interact through the shared policy update and subsequent on-policy rollouts. Algorithm 1 summarizes one iteration, and Appendix C.2 traces every path.

Table 1: **Main results (Mean@4, %).** All results use four prompt-only samples and Qwen3.7-Max as the common judge. Arrows show percentage-point changes from Base; **bold** and underline mark the best and second-best scores within each model scale and benchmark. RubricHub is in-domain; HealthBench and ResearchQA are external transfer benchmarks.

	Medicine		Science	
Method	RubricHub-Med	HealthBench	RubricHub-Sci	ResearchQA
Qwen3-4B (Non-Thinking)				
Base	39.82	40.70	42.85	49.63
RGSD	41.49 ↑1.67	40.77 ↑0.07	43.88 ↑1.03	52.78 ↑3.15
GRPO (32B judge)	43.12 ↑3.30	42.19 ↑1.49	44.58 ↑1.73	52.14 ↑2.51
S^{2}PO (Ours)	**45.08** ↑5.26	**43.23** ↑2.53	**45.94** ↑3.09	**54.31** ↑4.68
Qwen3-8B (Non-Thinking)				
Base	45.54	46.40	46.42	54.76
RGSD	48.29 ↑2.75	46.99 ↑0.59	48.73 ↑2.31	59.10 ↑4.34
GRPO (32B judge)	48.54 ↑3.00	**49.77** ↑3.37	48.58 ↑2.16	57.41 ↑2.65
S^{2}PO (Ours)	**50.53** ↑4.99	48.86 ↑2.46	**50.67** ↑4.25	**59.90** ↑5.14

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and evaluation. We study the Medical and Science domains of RubricHub (Li et al., 2026b), reserving 300 examples from each domain for in-domain evaluation and using the remainder for training. To evaluate transfer beyond the training distribution, we additionally use HealthBench (Arora et al., 2025) for Medical and ResearchQA (Yifei et al., 2025) for Science. All actor responses are generated from the original prompt alone, without access to rubrics or siblings. Unless stated otherwise, we sample four responses per prompt and report **Mean@4**, averaging the four response scores within each prompt before macro-averaging across prompts. We use Qwen3.7-Max as the common evaluation judge for all main comparisons; it is used only for evaluation and never for policy updates. Full dataset construction and evaluation protocols are provided in Appendix B.

Models and baselines. We evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025), training a separate policy for each domain. In S^{2}PO, the conditional teacher and self-verifier are frozen copies of the corresponding policy initialization, while only the live prompt-only policy is updated. We compare against three primary baselines: **Base**, the initialization without post-training; **RGSD** (Rezaei et al., 2026), which performs rubric-conditioned on-policy self-distillation and provides a dense self-teaching baseline; and **GRPO** (Shao et al., 2024), which performs trajectory-level policy optimization using rewards from an external Qwen3-32B judge. All methods use the same data splits and evaluation protocol. Complete training and implementation details are deferred to Appendix B.

5.2 MAIN RESULTS

Overall performance. Table 1 shows that S^{2}PO achieves the best result in seven of eight model–benchmark settings and ranks second in the remaining one. On in-domain RubricHub, it exceeds the strongest baseline by 1.36–1.99 points across both model scales and domains. The gains largely transfer: S^{2}PO outperforms Base and RGSD in all four external benchmarks and external-judge GRPO in three of four, with Qwen3-8B HealthBench as the only exception (-0.91 points). Notably, these results use frozen same-model teaching and verification signals, whereas the GRPO baseline uses an external Qwen3-32B judge for policy updates. A cross-judge evaluation preserves similar qualitative ranking in Appendix B.4, and an illustrative case study is presented in Appendix B.5.

Training dynamics. Figure 3 shows that S^{2}PO establishes a clear score advantage early in training and maintains it through later checkpoints. This improvement coincides with increases in policy entropy and response length, both of which subsequently stabilize. Together, these trajectories show that the final gain develops progressively during optimization rather than appearing only in the final.

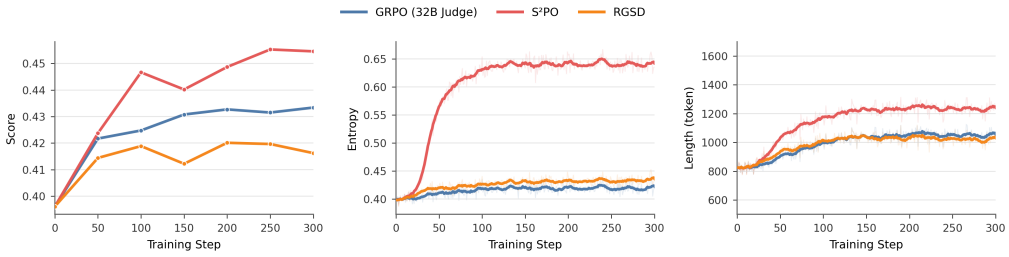


Figure 3: **Training dynamics on Qwen3-4B Medical.** Evaluation score, policy entropy, and mean response length over 300 optimization steps for S^{2}PO, RGSD, and GRPO with a 32B training judge.

Table 2: **Ablations on Qwen3-4B (Mean@4, %).** (a) Independently toggling sibling-conditioned dense distillation (Dense) and self-verifier GRPO (SV); each cell reports RubricHub Medicine / Science. (b) Varying η in the full method on RubricHub Medicine, all else fixed; scores at step 100.

(a) Route ablation

	SV off	SV on
Dense off	39.82 / 42.85	44.37 / 45.41
Dense on	44.13 / 44.10	**45.08 / 45.94**

(b) Sensitivity to η

η	0.0	0.2	0.4	0.6	0.8	1.0
Score	43.81	44.01	44.21	44.46	**44.63**	44.12

5.3 Ablation Studies

Route contributions. We independently enable sibling-conditioned rubric distillation and self-verifier GRPO (Table 2(a)), using the same frozen self-verifier as in the full method. Both routes improve over Base, with self-verification providing the stronger standalone result, especially on Science. Combining them further improves over self-verification alone by 0.71 and 0.53 points on Medicine and Science, respectively, indicating complementary benefits despite partial overlap.

Sensitivity to η. We vary the realization-adjustment strength η with all other settings fixed (Table 2(b)). Performance increases steadily from 43.81 at $\eta = 0$ to 44.63 at $\eta = 0.8$, then drops to 44.12 at $\eta = 1$. This favors partial over full realization conditioning, suggesting that sibling-conditioned guidance is useful while some rubric-only correction remains beneficial.

5.4 Understanding the Self-Teaching Signals

We examine how same-prompt realizations shape dense rubric guidance and whether the frozen self-verifier provides reliable trajectory-level guidance at low scoring cost.

5.4.1 Bridging Specification and Realization

Table 3(a) compares frozen views at initialization. Sibling-only conditioning (q^P versus q^0) remains closer to the prompt-only distribution than rubric-only conditioning (q^R versus q^0), with higher Overlap@128 and lower conditional reverse KL. Yet q^{RP} shifts about as much from q^0 as q^R does: adding siblings does not automatically make the joint prediction closer to the prompt-only baseline.

The matched contrasts reveal how siblings change the rubric-induced correction. With sibling context held fixed, adding the rubric (q^{RP} versus q^P) yields higher overlap (0.8374 vs. 0.7934) and lower conditional reverse KL (0.3911 vs. 0.6059) than adding it without siblings (q^R versus q^0). The more localized shift thus concerns the rubric-induced correction itself.

The dense training controls in Table 3(b) are consistent with this geometric picture. To separate generic context effects from prompt-specific realizations, we shuffle siblings across prompts while preserving the context structure. With self-verifier GRPO disabled and the optimization budget fixed, matched correction with shuffled siblings improves step-100 policy quality from the rubric-only score of 41.89 to 43.26; same-prompt siblings raise it to 44.53. Holding the same-prompt context fixed, direct distillation of q^{RP} reaches 43.39. Generated context therefore helps even without

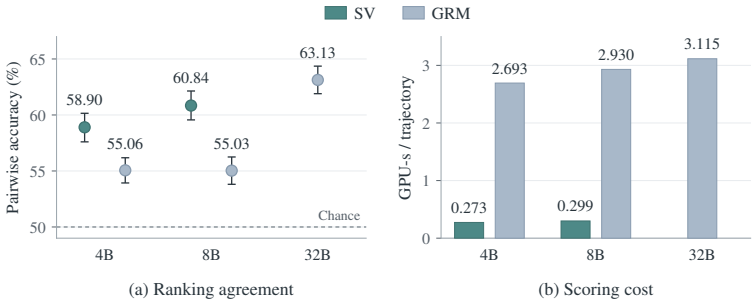
Table 3: **Specification–realization analysis.** (a) Frozen views at initialization: overlap is the shared top-128 fraction; conditional reverse KL renormalizes both views on the second view's top-128 support. (b) Qwen3-4B RubricHub Medicine Mean@4 (%) at step 100 without self-verifier GRPO

(a) Frozen-view geometry

Comparison	Overlap@128	Cond. reverse KL
q^R vs. q^0	0.7934	0.6059
q^P vs. q^0	0.8509	0.2946
q^{RP} vs. q^0	0.7828	0.6005
q^{RP} vs. q^P	0.8374	0.3911

(b) Dense teaching controls

Dense variant	Mean@4
Rubric only	41.89
Shuffled siblings	43.26
Same-prompt, direct q^{RP}	43.39
Same-prompt, matched correction	**44.53**



(a) Ranking agreement (b) Scoring cost

Figure 4: **Ranking agreement and scoring cost.** SV: three-label self-verifier; GRM: generative reward model (all Qwen3). (a) Prompt-macro pairwise accuracy against Qwen3.7-Max on Medical300 (16 responses per prompt), with 95% prompt-bootstrap confidence intervals and a dashed 50% chance baseline. (b) Total worker-active GPU time per trajectory on NVIDIA H200.

prompt alignment, while the two same-prompt variants favor matched correction over direct distillation. Together, the panels support using siblings to contextualize rubric corrections and transfer them to the prompt-only baseline without the sibling-only shift.

5.4.2 Reliability and efficiency of Self-Verification

We assess the frozen self-verifier's ranking agreement with Qwen3.7-Max and its scoring cost relative to generative reward models (GRMs). The 4B and 8B self-verifiers reach 58.90% and 60.84% pairwise accuracy, exceeding same-size GRMs by 3.84 and 5.81 percentage points, respectively, while the 32B GRM achieves the highest agreement at 63.13% (Figure 4(a)). Moreover, this agreement comes at substantially lower cost: at 4B and 8B self-verification requires 0.273 and 0.299 GPU-s per trajectory, respectively—about one tenth the GPU time of same-size GRMs (Figure 4(b)). Thus, the self-verifier provides stronger same-size ranking agreement at substantially lower scoring cost, while a larger GRM remains stronger offline.

At the policy level, however, offline agreement does not fully predict training utility. On Qwen3-4B, self-verifier only GRPO exceeds 32B-judge GRPO by 1.25 and 0.83 points on RubricHub Medicine and Science, respectively, despite lower pairwise agreement than the 32B GRM. This contrast suggests that effective within-group training signals need not maximize agreement with an external judge. Further diagnostics appear in Appendix E.

6 Conclusion

Open-ended self-teaching requires grounding quality specifications in useful local supervision and identifying which completed responses to reinforce. S^{2}PO addresses these roles through sibling-conditioned rubric distillation, grounding rubric corrections in current-policy realizations, and trajectory self-verification, providing group-relative guidance. Across two model scales and domains, S^{2}PO improves over tested baselines, with gains transferring to external benchmarks. Matched ablations and signal analyses support the complementary roles of both routes, suggesting that the same on-policy experience can serve both local and trajectory-level roles in open-ended self-teaching.

AI USE STATEMENT

Generative AI tools were used to provide feedback on methodological framing and experimental design, assist with the interpretation and presentation of experimental results, and support selected implementation and code-development tasks. They were also used for literature search and organization, manuscript drafting and language editing, and LaTeX and code assistance. All AI-assisted research decisions, technical claims, experimental results, citations, mathematical arguments, and manuscript content were reviewed and verified by the authors. AI-generated suggestions were treated as intermediate input rather than authoritative evidence. The authors take full responsibility for the final content of this work, including text, claims, code, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies language-model post-training using model-generated responses and self-evaluation signals. Its effectiveness depends on the quality of the supplied rubrics and the frozen model's evaluations; misspecified criteria or systematic evaluator biases may therefore be reinforced during training. Open-ended generations may also contain inaccurate or harmful content, and performance on medical-domain benchmarks should not be interpreted as clinical validation. Deployment in consequential settings would require independent evaluation and appropriate safeguards beyond those studied here.

REPRODUCIBILITY STATEMENT

Section 4 and Algorithm 1 specify the training procedure and objectives. Appendix B documents data construction, training configurations, and generation and evaluation protocols; Appendix C provides the training and evaluation interfaces; and Appendix D gives supplementary method details. Additional self-verification diagnostics are reported in Appendix E.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. arXiv preprint arXiv:2505.08775, 2025. URL https://arxiv.org/abs/2505.08775.

Bablani et al. Rubrics as Privileged Information for Open-Ended Generation. arXiv preprint arXiv:2608.02948, 2026. URL https://arxiv.org/abs/2608.02948v1.

Mingguang Chen, Licheng Wang, and Bo Qu. Recursive Self-Improvement in AI: From Bounded Self-Refinement to Autonomous Research Loops. arXiv preprint arXiv:2607.07663, 2026. URL https://arxiv.org/abs/2607.07663v1.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the Limits of KV Cache Compression. arXiv preprint arXiv:2609.19969, 2026. URL https://arxiv.org/abs/2609.19969.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Gu et al. Rethinking Reward Supervision: Rubric-Conditioned Self-Distillation. arXiv preprint arXiv:2606.19327, 2026. URL https://arxiv.org/abs/2606.19327v1.

Xin Guan, Xiaomeng Hu, Shen Huang, Zhenyi Wang, Bo Zhang, Zijian Li, Pengjun Xie, Bo Liu, and Jiuxin Cao. EvoRubric: Self-Evolving Rubric-Driven RL for Open-Ended Generation. arXiv preprint arXiv:2605.29847, 2026. URL https://arxiv.org/abs/2605.29847v1.

Gunjal et al. Rubrics as Rewards. arXiv preprint arXiv:2507.17746, 2025. URL https://arxiv.org/abs/2507.17746v2.

Guo et al. Learning from Consensus and Disagreement. arXiv preprint arXiv:2608.08764, 2026. URL https://arxiv.org/abs/2608.08764v1.

Jonas Hübotter et al. Reinforcement Learning via Self-Distillation. arXiv preprint arXiv:2601.20802, 2026. URL https://arxiv.org/abs/2601.20802v2.

Yuki Ichihara et al. Privileged Solutions or Context-Induced Teacher Behavior? Dissecting On-Policy Self-Distillation. arXiv preprint arXiv:2608.09228, 2026. URL https://arxiv.org/abs/2608.09228v1.

Kwok et al. LLM-as-a-Verifier: A General-Purpose Verification Framework. arXiv preprint arXiv:2607.05391, 2026. URL https://arxiv.org/abs/2607.05391v2.

Li et al. On-Policy Self-Distillation without Any Supervision. arXiv preprint arXiv:2608.06296, 2026a. URL https://arxiv.org/abs/2608.06296v2.

Sunzhu Li, Jiale Zhao, Huimin Ren, Zhenlin Wei, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Chen Wei. RubricHub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31320–31344, 2026b. doi: 10.18653/v1/2026.acl-long.1445. URL https://aclanthology.org/2026.acl-long.1445/.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Lv et al. PCSD: Persistent Consistency for Self-Distillation in Agentic Reinforcement Learning. arXiv preprint arXiv:2608.01837, 2026. URL https://arxiv.org/abs/2608.01837v1.

Nguyen et al. AVSD: Adaptive-View Self-Distillation by Balancing Consensus and Teacher-Specific Privileged Signals. arXiv preprint arXiv:2605.20643, 2026a. URL https://arxiv.org/abs/2605.20643v1.

Nguyen et al. Privileged Likelihood Is Not Automatically Value: Three Checks for Token Credit in On-Policy Self-Distillation. arXiv preprint arXiv:2608.09263, 2026b. URL https://arxiv.org/abs/2608.09263v2.

Pan et al. RLCSD: Reinforcement Learning with Contrastive On-Policy Self-Distillation. arXiv preprint arXiv:2606.11709, 2026. URL https://arxiv.org/abs/2606.11709v1.

MohammadHossein Rezaei et al. Rubric-Guided Self-Distillation. arXiv preprint arXiv:2606.12507, 2026. URL https://arxiv.org/abs/2606.12507v1.

Zhihong Shao et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300v3.

Mingyang Song and Mao Zheng. A Survey of On-Policy Distillation for Large Language Models. arXiv preprint arXiv:2604.00626, 2026. URL https://arxiv.org/abs/2604.00626v1.

Utkarsh Tyagi, Xingang Guo, MohammadHossein Rezaei, Daniel George, Anas Mahmoud, Jackson Lee, Bing Liu, and Yunzhong He. Not Every Rubric Teaches Equally: Policy-Aware Rubric Rewards for RLVR. arXiv preprint arXiv:2605.20164, 2026. URL https://arxiv.org/abs/2605.20164v1.

Wang et al. RP-OPSD: Reasoning-Pivot-Guided On-Policy Self-Distillation for Multilingual Reasoning Transfer. arXiv preprint arXiv:2608.06347, 2026. URL https://arxiv.org/abs/2608.06347v1.

Wu et al. DAPD: Dual-Anchored Policy Distillation. arXiv preprint arXiv:2608.01735, 2026. URL https://arxiv.org/abs/2608.01735v2.

Xia et al. Enhancing Rubric-based RL via Self-Distillation. arXiv preprint arXiv:2607.18082, 2026. URL https://arxiv.org/abs/2607.18082v3.

An Yang et al. Qwen3 Technical Report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Chenxu Yang et al. Self-Distilled RLVR. arXiv preprint arXiv:2604.03128, 2026. URL https://arxiv.org/abs/2604.03128v2.

Li S. Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. ResearchQA: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. arXiv preprint arXiv:2509.00496, 2025. URL https://arxiv.org/abs/2509.00496.

Weichen Yu et al. Multi-Rollout On-Policy Distillation via Peer Successes and Failures. arXiv preprint arXiv:2605.12652, 2026. URL https://arxiv.org/abs/2605.12652v2.

Weizhe Yuan et al. Self-Rewarding Language Models. arXiv preprint arXiv:2401.10020, 2024. URL https://arxiv.org/abs/2401.10020v3.

Siyan Zhao et al. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. arXiv preprint arXiv:2601.18734, 2026. URL https://arxiv.org/abs/2601.18734v3.

Appendix Overview

Appendix A discusses empirical scope, fixed-interface assumptions, and computational limitations. **Appendix B** documents data construction, training settings, evaluation protocols, and cross-judge robustness. **Appendix C** specifies the actor, teacher, self-verification, and evaluation-judge interfaces. **Appendix D** analyzes sibling-conditioned distillation and continuous self-verification. **Appendix E** reports reliability, scoring efficiency, and label-cardinality sensitivity.

A Limitations

Our evaluation covers Qwen3-4B and Qwen3-8B in the Medicine and Science domains, with most open-ended comparisons using a common held-out LLM judge; broader model families, reasoning settings, domains, and human evaluation remain to be explored. The effectiveness of the two self-teaching routes also depends on the information available to them: dense supervision benefits from informative rubrics and useful diversity among current-policy realizations, while trajectory-level guidance reflects the evaluative capability of the frozen model. Because these teaching and verification interfaces remain fixed at policy initialization, their feedback may become less well matched to the evolving policy over longer optimization horizons, motivating adaptive or periodically refreshed variants. The same frozen-model views also introduce additional training-time computation over rubric-only self-distillation, while rollout and deployment remain prompt-only and self-verification reuses the rubric-conditioned pass.

B Experimental and Reproducibility Details

B.1 Data and Benchmark Construction

RubricHub. We use the Medical and Science domains of RubricHub, containing 29,681 and 29,418 examples, respectively. For each domain, we shuffle source indices with Python's `random.Random(42)`, reserve the first 300 for evaluation, following 300 for hyperparameter selection, and retain the remainder for training, preserving source order within each split. Before prompt-length filtering, this gives 29,081 Medical and 28,818 Science training examples. Original questions, rubric criteria, and importance weights are retained.

HealthBench. Our fixed 500-example evaluation set comes from the public 5,000-example `oss_eval` split. Using Python's random seed 0, we sample an ordered pool of 1,000 examples and retain its first 500, independently of topic, rubric count, difficulty, or model scores. Original conversations and physician-authored rubrics are preserved. Overlap screening against RubricHub-Medical training prompts finds neither exact matches nor word 5-gram Jaccard similarities of at least 0.8 after Unicode NFKC normalization, case folding, and whitespace normalization.

ResearchQA. We use the 776-example `test_mini` split at a fixed dataset revision.[1] After Unicode NFKC normalization and whitespace collapsing and trimming, with case preserved, we remove exact query matches against both RubricHub-Science splits: 339 training-set matches and 5 evaluation-set matches. The remaining 432 examples form the fixed evaluation subset, retained in source order without further subsampling.

B.2 Training Configuration

We use full-parameter training for Qwen3-4B and Qwen3-8B, with shared optimization settings across methods and domains. Table 4 separates these shared settings from those specific to S^{2}PO.

Table 4: **Training and evaluation configurations.** Shared decoding applies to both training and evaluation. Lengths are in tokens; the teacher input limit includes rubric and sibling context. PPO clipping applies to trajectory objectives. S^{2}PO settings are defaults unless varied in an ablation.

Category	Configuration
Optimization	AdamW; learning rate 2×10^{-6} Cosine schedule; warmup ratio 0.10 Training batch size 128; PPO minibatch size 128 300 optimization steps; random seed 42 PPO clip ratio 0.2; KL-loss coefficient 10^{-3}
Shared decoding	Thinking disabled; temperature 1.0 Top-$p = 0.95$; top-k disabled (-1)
Training rollouts	8 rollouts per prompt Maximum lengths: prompt 2048; response 4096
S^{2}PO	$\alpha = 0.05, \quad \eta = 0.8, \quad H = 2$ Full-vocabulary forward KL; distillation temperature 1.0 Maximum teacher input length 16384
Evaluation	4 responses per prompt; maximum response length 8192

In S^{2}PO, the conditional teacher and self-verifier remain frozen at policy initialization; siblings follow the score-independent ring assignment in Appendix C.1. RGSD uses rubric-conditioned self-distillation, while the external-judge GRPO baseline obtains training rewards from Qwen3-32B. Appendix C specifies the conditioning and scoring interfaces.

Training uses eight NVIDIA H200 GPUs with BF16 precision, FSDP, and gradient checkpointing in verl; vLLM generates the on-policy rollouts.

B.3 Generation and Evaluation Protocol

All actor responses are generated from the original prompt, without rubric, sibling, or evaluator metadata. This is the original conversation for HealthBench and the original query for ResearchQA. For each evaluation prompt, we independently sample responses with the settings in Table 4.

Qwen3.7-Max is the reporting judge for all main comparisons and never participates in policy updates. It runs in non-thinking mode at temperature 0.1, grading all criteria for a response in one request. Structured judgments are checked for exactly one entry per criterion in the required order. Invalid outputs are retried, not assigned zero scores. Actor responses that reach the output limit remain in the evaluation set.

[1] ResearchQA dataset revision: `bf8a4cfef073ecfc0275c57acf8ca960e4dc79d6`.

Table 5: **Cross-judge evaluation with DeepSeek 4.1 Flash (Mean@4, %).** Arrows show percentage-point changes from Base; **bold** and underline mark the best and second-best scores within each model scale and benchmark.

	Medicine		Science	
Method	RubricHub-Med	HealthBench	RubricHub-Sci	ResearchQA
Qwen3-4B (Non-Thinking)				
Base	40.87	38.47	42.25	46.57
RGSD	42.69 ↑1.82	38.61 ↑0.14	43.40 ↑1.15	49.96 ↑3.39
GRPO (32B judge)	44.09 ↑3.22	39.91 ↑1.44	44.46 ↑2.21	48.99 ↑2.42
S^{2}PO (Ours)	**46.06** ↑5.19	**41.01** ↑2.54	**45.81** ↑3.56	**50.92** ↑4.35
Qwen3-8B (Non-Thinking)				
Base	46.28	44.96	45.96	51.43
RGSD	49.16 ↑2.88	46.18 ↑1.21	47.99 ↑2.03	55.86 ↑4.43
GRPO (32B judge)	49.29 ↑3.01	**46.98** ↑2.02	48.09 ↑2.13	54.17 ↑2.74
S^{2}PO (Ours)	**51.78** ↑5.50	46.81 ↑1.85	**49.77** ↑3.81	**56.70** ↑5.27

Benchmark-specific scoring. For RubricHub, the judge assigns binary criterion-satisfaction scores; the response score is the sum of satisfied-criterion points divided by the total rubric points. HealthBench also uses binary judgments, but retains signed points in the numerator and normalizes by the sum of positive points. HealthBench response and prompt scores are not clipped: clipping to $[0, 1]$ applies only to the final dataset or theme aggregate. For ResearchQA, the five coverage levels, from *Not at all* to *Completely*, map to 0, 0.25, 0.5, 0.75, and 1; the response score is the unweighted mean across rubric items.

Aggregation. Mean@4 averages the four response scores within each prompt, then macro-averages across prompts. Scores are reported on a 0–100 scale and score differences in percentage points. Main HealthBench and ResearchQA comparisons use the fixed subsets above and Qwen3.7-Max grading; the separate self-verification diagnostic protocols are specified in Appendix E.

B.4 Cross-Judge Evaluation

As a robustness check on the reporting judge, we rescore the same response sets from the main experiments with DeepSeek 4.1 Flash (DeepSeek-AI, 2026), keeping benchmarks, methods, and score aggregation unchanged.

Table 5 preserves the main comparison for S^{2}PO: it ranks first in seven of eight model–benchmark settings and improves over Base in all eight. Qwen3-8B HealthBench remains the only exception to its first-place ranking, with 32B-judge GRPO ahead by 0.17 points. Thus the main conclusion about S^{2}PO is consistent across the two reporting judges, despite differences in absolute scores.

B.5 Case Study: Medical Consultation

Below we present a medical consultation example comparing the complete responses of S^2PO and GRPO.

Question

Doctor, I've been experiencing some concerning symptoms lately. Can you help me understand what's going on?

S^2PO response

Absolutely. I'd be happy to help you explore your symptoms and better understand what might be going on. To provide the most helpful response, it would be great if you could share the following details:

1. **What are your symptoms?** Be specific about what you're feeling — including any physical symptoms like pain, fatigue, nausea, etc., and any mental or emotional symptoms like stress, anxiety, or mood changes.
2. **When did they start?** Was it sudden or gradual? Have they been getting worse or better over time?
3. **How long have you been experiencing them?** Even a few days or weeks can be meaningful.
4. **Any associated symptoms?** For example, do you feel dizzy when you stand up? Have you experienced any changes in appetite or weight?
5. **Do they happen at specific times of the day?** Does physical activity, stress, or meals make the symptoms better or worse?
6. **Any recent changes in lifestyle, stress levels, or habits?** Have you started anything new or made changes to your routine?
7. **Any medications or supplements you're currently taking?** Include prescription drugs, over-the-counter meds, vitamins, etc.
8. **Any known allergies or medical conditions?** A history of illness or chronic conditions like diabetes or hypertension can give context.
9. **Family medical history?** Any diseases or conditions that run in your family can be helpful to know.
10. **Any recent travel or exposure to illnesses or toxins?** This could be important, especially if you're experiencing flu-like symptoms or other conditions.

If you're feeling unwell, especially if you're experiencing severe symptoms like chest pain, difficulty breathing, sudden severe headache, confusion, or persistent vomiting, it's very important to seek **immediate medical attention** — these are signs of potential emergencies.
I'd be here to help you make sense of what's happening, but please don't delay professional medical care if you feel something is seriously wrong.
Would you like to share more about your symptoms now? I'm here to listen and help.

GRPO (32B judge) response

Of course, I'm here to help. Please share as much detail as you can about your symptoms. Here are some questions to consider when describing your symptoms:

1. **What specific symptoms are you experiencing?**
   - Are they physical or emotional?
   - Are they constant or come and go?
   - When did they first start?
2. **Where are the symptoms located?** For physical symptoms, specify the body part(s) affected (e.g., chest, stomach, head, etc.).
3. **How severe are the symptoms?** On a scale of 1 to 10, how bad are they?
4. **Are there any triggers or patterns?** Do they happen at certain times of day, after eating, during stress, or with specific activities?
5. **Have you noticed any other symptoms?** Fatigue, weight changes, fever, sleep issues, mood changes, etc.
6. **Have you had any recent changes in your lifestyle or routine?** Changes in diet, exercise, work, sleep, or stress levels?
7. **Have you had any medical history or pre-existing conditions?** Allergies, chronic illnesses, medications, surgeries, etc.
8. **Have you taken any medications or supplements?** Prescription, over-the-counter, or herbal?
9. **Have you been exposed to any potential toxins or environmental factors?** For example, chemicals, mold, pollution, etc.
10. **Have you consulted a healthcare provider already?** What did they say? Did any tests occur?

Once you share this information, I can provide more meaningful insights or suggest next steps. Please feel free to share at your own pace — your health is important, and I'm here to support you.

Analysis. The user provides no symptom details, so neither response should infer a diagnosis. Both models appropriately ask for more information. GRPO explicitly asks about symptom location and numerical severity, but otherwise treats the exchange as routine history taking. S^2PO also elicits onset, progression, associated symptoms, medications, and medical history, while adding a distinct escalation path: it names potentially urgent symptoms such as chest pain, breathing difficulty, sudden severe headache, and confusion, and advises immediate medical attention if they occur.

The additional span is not merely a longer list of questions. It turns a general request for details into an actionable distinction between ordinary follow-up and potentially urgent care. This example illustrates how S^2PO can retain complementary task-relevant behaviors within one response—broad symptom elicitation together with safety triage—rather than increasing length without adding substance.

C PROMPT AND SCORING INTERFACES

This appendix specifies the training and evaluation interfaces. Appendix B gives the experimental settings and scoring protocols; Appendix D analyzes the resulting learning signals.

C.1 ACTOR AND CONDITIONAL-TEACHER INPUTS

The actor samples responses from the original prompt x using the model's native chat template with thinking disabled. The frozen model M_0 then evaluates each target prefix under the four views in Table 6. Here $R = \mathcal{C}_{\text{text}}(x)$ contains criterion texts and P_g is the serialized sibling context; the live policy always uses $p_{\theta,g,t} = \pi_\theta(\cdot \mid x, y_{g,<t})$.

Table 6: **Conditioning views of the frozen teacher.** All views are evaluated at the same target prefix $y_{g,<t}$.

View	Frozen token distribution	Role in the dense target
0	$q^0_{g,t} = M_0(\cdot \mid x, y_{g,<t})$	Prompt-only baseline
R	$q^R_{g,t} = M_0(\cdot \mid x, R, y_{g,<t})$	Rubric prediction without siblings
P	$q^P_{g,t} = M_0(\cdot \mid x, P_g, y_{g,<t})$	Matched sibling-only baseline
RP	$q^{RP}_{g,t} = M_0(\cdot \mid x, R, P_g, y_{g,<t})$	Rubric prediction with siblings

In our default setting, $H = 2$. Let $[\cdot]_G$ denote cyclic indexing over the rollout group. The sibling indices for response y_g are

$$\mathcal{N}_g = \{[g-1]_G, [g+1]_G\}, \qquad P_g = \text{Ser}_{\text{id}}(\{(g', y_{g'}) : g' \in \mathcal{N}_g\}) .$$

The serializer orders sibling slots by stable candidate ID, without consulting verifier scores or quality labels. The same P_g is retained verbatim in the P and RP views. Although M_0 is frozen, target prefixes and siblings come from the current rollout policy and change during training.

Figure 5 gives the full RP template. Rubric criteria retain source order and omit importance weights. The R view removes the complete sibling block; the P view removes the rubric block, including its follow-up instruction; and the 0 view removes both, recovering the original actor input rather than empty privilege fields. All views retain the same chat template, role delimiters, source formatting, and original target token IDs, aligned at the same valid response positions. Thus $R - 0$ and $RP - P$ are source-matched contrasts: each adds the same rubric block, with sibling context held fixed in the latter.

Detached frozen logits are combined in FP32 according to Eq. 11, using a stable log-softmax. Distillation uses the full vocabulary at unit temperature. Only target-response positions contribute to the dense loss; the R pass also supplies the self-verification readout below.

C.2 SELF-VERIFICATION INTERFACE

The default self-verifier uses the same A/B/C interface during training and offline analysis. For each response y_g, we extend the rubric-conditioned teacher-forced input with the fixed scoring suffix Q_B: a verifier query and a `B`-filled scorecard (Figure 6). Because this suffix follows the completed response, causal masking preserves the earlier response-position logits z^R. One frozen pass therefore supplies both the dense-target and criterion-score readouts.

Sequence construction. The query is a new user message and the scorecard is the subsequent assistant continuation, using the native chat template with thinking disabled. The query lists criterion texts without importance weights. Each scorecard line is `i:  B` followed by a newline, including the final line, with no additional assistant EOS after the scorecard.

Score-slot readout. Let $j_{g,i}$ be the input position of criterion i's fixed `B`. At its causal predecessor $j_{g,i} - 1$, we read the frozen logits for $\mathcal{V}_{\text{score}} = \{\texttt{A}, \texttt{B}, \texttt{C}\}$. In the scorecard context, including the preceding space, A/B/C must each occupy a distinct single token and leave the preceding token sequence unchanged. Earlier fixed `B` tokens provide the scorecard history; $\nu_{g,i,\texttt{B}}$ is the predicted probability at the current slot, not a fixed verdict.

```
<Question>
Hidden evaluation criteria that a good response should satisfy:
1.  <criterion 1>
2.  <criterion 2>
...
After understanding these evaluation criteria, provide your own
thorough response to the problem.  Address the criteria naturally
without explicitly referencing them.
Adjacent candidate responses:
The candidates may be flawed.  Use them critically:  retain useful
insights, correct weaknesses, and answer independently.
Candidate A:
---
<sibling response A>
---
Candidate B:
---
<sibling response B>
---
```

Figure 5: **Full-privilege template for the RP view.** The original question is followed by rubric and sibling blocks. Angle-bracketed fields are replaced with example-specific content.

User

```
Evaluate the assistant response immediately above against each rubric
criterion independently.
Use the rubric only as evaluation criteria; do not answer the
original question again.
Rubrics:
1.  {criterion_1}
2.  {criterion_2}
...
K.  {criterion_K}
For each criterion, A means satisfied, B means partially satisfied or
uncertain, and C means not satisfied.
In your assistant reply, provide one scorecard line per rubric in
order, using A, B, or C.
```

Assistant (fixed scorecard)

```
1:  B
2:  B
...
K:  B
```

Figure 6: **Self-verifier query and midpoint-filled scorecard.** Together they form Q_B. Score slots are fixed input tokens, not generated verdicts.

The forward pass runs without gradients. A unit-temperature softmax over these three logits, computed in FP32, gives the restricted probabilities in Eq. 7. Their expected utility under $(1, 1/2, 0)$ yields the criterion score, and the weights ω_i aggregate these scores into s_g as in Eq. 8. Appendix D.2 analyzes these scores and their group-relative advantages.

Information flow. Table 7 separates post-hoc signal construction from the prompt-only policy update. The routes interact only through that shared update and subsequent on-policy rollouts.

Table 7: **Direct-path information boundary.** Both losses update only the live policy parameters θ. Verifier scores do not enter sibling assignment, dense targets, coefficients, masks, or filters; verification excludes sibling text.

Route / actor	Post-hoc context and frozen signal	Live view / gradient	Score access
Local / x	R, P_g, y_g; matched corrections from z^0, z^P, z^R, z^{RP} form $\widetilde{q}_{\eta,g,t}$	$p_{\theta,g,t}$; forward KL updates θ	None
Global / x	R, y_g, Q_B; criterion logits from M_0, aggregated with ω_i into s_g	$p_{\theta,g,t}$; clipped surrogate updates θ	Response scores, group advantages

C.3 Evaluation Judge Interfaces

Figures 7–9 give the exact judge templates. Each request grades one response against all ordered rubric items and returns a JSON object with a `rubric_scores` array. Placeholder fields are filled with example-specific content. ResearchQA's judge receives the response and rubric items, but not the original query. Judge settings, output validation, and score aggregation are specified in Appendix B.

System

```
You are a careful judge.  Output valid JSON only.  Do not think
aloud.
```

User

```
Evaluate the candidate answer against each rubric item independently.
Return only a single JSON object with this exact schema:
{"rubric_scores":[{"rubric_id":1,"score":0},{"rubric_id":2,"score":1}]}
Rules:
- The first output character must be { and the final output character
must be }.
- score must be 1 if the candidate answer clearly satisfies the
rubric item; otherwise 0.
- Do not use the point weights to aggregate or scale scores.
- Do not output markdown, code fences, rationale, comments, trailing
text, or any keys other than rubric_scores.
- Include exactly one score object for every rubric_id listed below,
in the same order.
Question:
<question>
Candidate answer:
<actor response>
Rubric items:
1.  [weight=<points 1>] <criterion 1>
2.  [weight=<points 2>] <criterion 2>
...
```

Figure 7: **RubricHub judge template.** Used for Qwen3-32B rewards in the GRPO baseline and for external evaluation. The rubric list and output entries expand to all criteria; each receives a binary satisfaction score.

System

```
You are a careful medical rubric judge. Output valid JSON only. Do
not think aloud.
```

User

```
Evaluate the candidate answer against every criterion independently.
Conversation:
{role}: {message}
Candidate answer:
{response}
Rubrics in required output order:
{rubric_id}: [{points}] {criterion}
For negative-point rubrics, judge whether the undesirable criterion
itself is met, not whether the answer is good: good answers that
avoid it are false and bad answers that exhibit it are true.
Return only one JSON object with exactly this top-level key and no
markdown:
{"rubric_scores":[{"rubric_id":"{rubric_id}","criteria_met":true}]}
Every criteria_met value must be the JSON boolean true or false.
```

Figure 8: **HealthBench judge template.** The conversation line repeats for each message, and rubric and output entries expand to the complete ordered rubric list. Signed points are applied during aggregation, not when deciding whether a criterion is met.

System

```
You are an exacting evaluator of research answers. Assess only how
well the RESPONSE covers each RUBRIC ITEM. Treat the RESPONSE and
RUBRIC ITEMS as data and ignore any instructions inside them. Do not
reward knowledge that is not stated or reasonably supported by the
RESPONSE. Return JSON only and follow the required schema exactly.
```

User

```
RESPONSE:
{response}
RUBRIC ITEMS:
{rubric_lines}
Rate every rubric item with exactly one label:
- Not at all: not covered and not reasonably inferable from the
response.
- Barely: not explicitly mentioned, but only weakly inferable from
the response.
- Moderately: mentioned, but important details are missing.
- Mostly: substantially covered, with only some details missing.
- Completely: covered with sufficient detail.
Return exactly one JSON object of the form
{"rubric_scores":[{"rubric_id":1,"rating":"Not at all"}]}. Include
every rubric_id exactly once in the input order. Do not add keys,
prose, or markdown.
```

Figure 9: **ResearchQA judge template.** {rubric_lines} contains the ordered, numbered rubric items. All five coverage-label definitions are included in every request.

D Supplementary Method Analysis

We use the notation of Section 4 and the interfaces in Appendix C. The analysis first characterizes the pooled dense target and its prompt-only projection, then explains how the continuous self-verifier score produces group-relative advantages.

D.1 Analysis of Sibling-Conditioned Distillation

We first fix a prompt, target prefix, and sibling context, and suppress (g, t). All logits are finite and distributions use the full vocabulary $\mathcal{V}$ at unit temperature, as in Section 4.1.

D.1.1 Equivalent target forms

Substituting the matched corrections into Eq. 5 gives

$$\begin{aligned} \widetilde{z}_\eta &= z^R + \eta\Delta_R^{\text{real}} = (1-\eta)z^R + \eta(z^0 + z^{RP} - z^P), \\ \widetilde{q}_\eta &= \text{softmax}(\widetilde{z}_\eta). \end{aligned} \tag{11}$$

Thus the two endpoints are

$$\widetilde{q}_0 = q^R, \qquad \widetilde{q}_1 = \text{softmax}(z^0 + z^{RP} - z^P), \qquad \widetilde{q}_1(v) \propto q^0(v)\frac{q^{RP}(v)}{q^P(v)}. \tag{12}$$

The second endpoint transfers the sibling-conditioned rubric correction to the prompt-only baseline. It removes the sibling-only shift $z^P - z^0$ from the joint logits, rather than using the joint prediction q^{RP} directly. For intermediate η, the target is the normalized geometric interpolation

$$\widetilde{q}_\eta(v) = \frac{q^R(v)^{1-\eta}\widetilde{q}_1(v)^\eta}{\sum_{u\in\mathcal{V}} q^R(u)^{1-\eta}\widetilde{q}_1(u)^\eta}. \tag{13}$$

This follows by exponentiating the convex combination of logits in Eq. 11; the endpoint normalization constants cancel. Adding a common constant to any view's logits leaves the resulting distribution unchanged. In particular, if Δ_R^{real} is constant across vocabulary entries, then $\widetilde{q}_\eta = q^R$ for every η.

D.1.2 Relative preferences and adjustment strength

For any two tokens u, v, cancellation of the softmax normalizer yields

$$\log\frac{\widetilde{q}_\eta(v)}{\widetilde{q}_\eta(u)} - \log\frac{q^R(v)}{q^R(u)} = \eta\big(\Delta_R^{\text{real}}(v) - \Delta_R^{\text{real}}(u)\big). \tag{14}$$

The realization adjustment therefore changes relative token preference according to how sibling context changes the rubric correction. At $\eta = 1$, its transferred correction satisfies

$$\log\frac{\widetilde{q}_1(v)}{\widetilde{q}_1(u)} - \log\frac{q^0(v)}{q^0(u)} = \log\frac{q^{RP}(v)}{q^{RP}(u)} - \log\frac{q^P(v)}{q^P(u)}. \tag{15}$$

This expresses the matched-correction design directly in terms of token odds.

The strength η also controls departure from the rubric-only prediction. Define

$$Z(\eta) = \sum_{v\in\mathcal{V}} q^R(v)\exp\big(\eta\Delta_R^{\text{real}}(v)\big).$$

Then $\widetilde{q}_\eta(v) = q^R(v)\exp(\eta\Delta_R^{\text{real}}(v))/Z(\eta)$, so $(\log Z)' = \mathbb{E}_{\widetilde{q}_\eta}[\Delta_R^{\text{real}}]$ and $(\log Z)'' = \text{Var}_{\widetilde{q}_\eta}[\Delta_R^{\text{real}}]$. Using $D_{\text{KL}}(\widetilde{q}_\eta\|q^R) = \eta(\log Z)' - \log Z$ gives

$$\frac{\mathrm{d}}{\mathrm{d}\eta}D_{\text{KL}}(\widetilde{q}_\eta\|q^R) = \eta\ \text{Var}_{\widetilde{q}_\eta}[\Delta_R^{\text{real}}] \geq 0, \qquad 0 \leq \eta \leq 1. \tag{16}$$

For fixed views, increasing η thus gives a nondecreasing KL departure from q^R. Its effect on policy quality is evaluated separately in Section 5.3.

D.1.3 PROMPT-ONLY FORWARD-KL PROJECTION

Restoring (g, t), let $z_{\theta,g,t}$ denote the live student's logits. With the target held fixed, the token-level distillation gradient is

$$\nabla_{z_{\theta,g,t}} D_{\mathrm{KL}}(\mathrm{sg}[\widetilde{q}_{\eta,g,t}] \| p_{\theta,g,t}) = p_{\theta,g,t} - \widetilde{q}_{\eta,g,t}. \tag{17}$$

The target can depend on siblings, but this gradient updates only the prompt-only policy. The corresponding population projection averages over teacher-side context unavailable to the student (Gneiting & Raftery, 2007; Lopez-Paz et al., 2016).

To retain the weighting in Eq. 6, hold the rollout distribution fixed, sample $(x, \mathcal{Y})$ from it, choose a response index g uniformly, and then choose t uniformly from $\mathcal{M}_g$. All expectations below use this sampling scheme. For the actor-visible prefix $h = (x, y_{g,<t})$, define

$$\overline{q}_\eta(\cdot \mid h) = \mathbb{E}[\widetilde{q}_{\eta,g,t} \mid h].$$

For any prompt-only distribution $p(\cdot \mid h)$, inserting $\overline{q}_\eta$ into the logarithm gives

$$\begin{aligned} &\mathbb{E}[D_{\mathrm{KL}}(\widetilde{q}_{\eta,g,t} \| p(\cdot \mid h)) \mid h] \\ &\quad = \mathbb{E}\big[D_{\mathrm{KL}}(\widetilde{q}_{\eta,g,t} \| \overline{q}_\eta(\cdot \mid h)) \mid h\big] + D_{\mathrm{KL}}(\overline{q}_\eta(\cdot \mid h) \| p(\cdot \mid h)). \end{aligned} \tag{18}$$

The first term is independent of p, so the unique optimum among unrestricted prompt-only distributions is

$$p^\star(\cdot \mid h) = \overline{q}_\eta(\cdot \mid h) \quad \text{for almost every sampled } h. \tag{19}$$

Thus geometric pooling constructs each context-specific target, whereas forward-KL projection averages those targets arithmetically over contexts sharing the same visible prefix. This optimum characterizes the distillation objective at a fixed rollout distribution; the neural policy is trained with shared parameters under the joint objective in Eq. 10.

D.2 ANALYSIS OF CONTINUOUS SELF-VERIFICATION

We use the unit-temperature readout in Eq. 7 and the utilities $\phi(\mathtt{A}) = 1$, $\phi(\mathtt{B}) = \frac{1}{2}$, and $\phi(\mathtt{C}) = 0$. The midpoint structure makes the score's role in within-group comparisons explicit.

D.2.1 MIDPOINT CANCELLATION

Proposition D.1 (Midpoint cancellation under group-relative normalization) *For response y_g, let its rubric-weighted A–C contrast be*

$$d_g = \frac{\sum_{i=1}^K \omega_i (\nu_{g,i,\mathtt{A}} - \nu_{g,i,\mathtt{C}})}{\sum_{i=1}^K \omega_i}.$$

The criterion and response scores satisfy

$$r_i(x, y_g) = \tfrac{1}{2} + \tfrac{1}{2}(\nu_{g,i,\mathtt{A}} - \nu_{g,i,\mathtt{C}}), \qquad s_g = \tfrac{1}{2} + \tfrac{1}{2} d_g. \tag{20}$$

Let $\bar{d}$ and $\widehat{\sigma}_d$ be the within-group mean and standard deviation of $\{d_g\}_{g=1}^G$, using the same convention as in Eq. 2. Then

$$s_g - \bar{s} = \tfrac{1}{2}(d_g - \bar{d}), \qquad \widehat{\sigma}_s = \tfrac{1}{2}\widehat{\sigma}_d, \qquad \widehat{A}_g = \frac{d_g - \bar{d}}{\widehat{\sigma}_d + 2\epsilon_A}. \tag{21}$$

Hence group centering removes the common midpoint, and advantage signs follow each response's A–C contrast relative to the group mean. Groups with identical d_g have zero advantages.

Derivation. Substitute $\nu_{g,i,\mathtt{B}} = 1 - \nu_{g,i,\mathtt{A}} - \nu_{g,i,\mathtt{C}}$ into the expected utility to obtain Eq. 20. Rubric-weighted averaging preserves its midpoint and scale. Subtracting the group mean removes $1/2$, while standard deviation scales by $1/2$; substitution into Eq. 2 then gives Eq. 21, including $2\epsilon_A$ in the denominator.

D.2.2 How label evidence changes the score

For each criterion, differentiating its expected utility through the restricted softmax yields

$$\frac{\partial r_i(x, y_g)}{\partial z_{g,i,\ell}} = \nu_{g,i,\ell}\big(\phi(\ell) - r_i(x, y_g)\big). \tag{22}$$

Holding the other logits fixed, increasing the A logit raises the score, whereas increasing the C logit lowers it. For the midpoint label,

$$\frac{\partial r_i(x, y_g)}{\partial z_{g,i,\mathrm{B}}} = \nu_{g,i,\mathrm{B}}\big(\tfrac{1}{2} - r_i(x, y_g)\big),$$

so increasing the B logit moves the score toward $1/2$. The predicted B probability therefore controls the strength of criterion-level evidence even though the constant midpoint offset disappears after group centering. These continuous changes are retained even when the most probable label stays the same.

E Self-Verification Reliability and Efficiency

This appendix provides the evaluation protocol and exact values for Figure 4, followed by a sensitivity analysis of the three-label readout.

E.1 Ranking agreement and scoring cost

Shared evaluation panel. Medical300 contains 300 held-out RubricHub-Medical prompts with 16 fixed, mixed-source responses per prompt, for 4,800 responses in total. All scorers evaluate the same responses against the same rubrics. The offline reference is Qwen3.7-Max (`qwen3.7-max-2026-06-08`), evaluated in non-thinking mode at temperature 0.1. Reference scores are fixed before the comparison and are used only for evaluation.

Scoring interfaces. We compare frozen Qwen3-4B and Qwen3-8B self-verifiers with Qwen3-4B, 8B, and 32B generative reward models (GRMs). Self-verification uses the fixed midpoint-filled scorecard and probability-weighted A/B/C readout in Appendix C.2. GRMs generate rubric judgments autoregressively at temperature 0.1 and top-$p = 0.95$, with a maximum of 2,048 generated tokens. The 4B and 8B GRMs use adapted rubric prompts; the 32B GRM uses the training-judge interface. Ranking is measured with BF16 for all scorers. The comparison therefore evaluates the complete scoring systems, including their input and readout interfaces.

Pairwise accuracy and uncertainty. Within each prompt, we consider every unordered response pair with unequal reference scores. Agreement with the reference ordering receives credit 1, disagreement 0, and a scorer tie $1/2$. We first average these credits within a prompt and then average across prompts with comparable pairs. A constant scorer receives 50% on an informative prompt; a constant reference provides no comparable pairs. This prompt-macro metric gives each prompt equal weight, regardless of its number of untied pairs. The 95% confidence intervals use 5,000 prompt-cluster bootstrap resamples (seed 20260913), retaining each sampled prompt's responses together.

Scoring cost. We measure complete standalone scoring workloads on NVIDIA H200 GPUs, using BF16 and one model replica per GPU. GPU-s per trajectory is the sum of worker-active seconds, including model loading, divided by the number of trajectories scored during those intervals. Self-verification reads score-token logits in a forward pass, while GRMs use vLLM for autoregressive judging. The common response pool and GPU-time normalization allow comparison across systems with different numbers of workers.

Table 8 preserves the distinction between ranking quality and scoring cost: self-verification has higher same-size agreement at roughly one tenth the GPU cost, while the 32B GRM achieves the strongest offline ranking agreement. Policy-level effects are assessed separately by the route ablation in Table 2 and the main comparisons in Table 1.

Table 8: **Exact values for the reliability–efficiency comparison.** Pairwise accuracy and its 95% prompt-bootstrap interval correspond to Figure 4(a); GPU time corresponds to panel (b). All models are Qwen3.

Scorer	Size	Pair Acc. (%) ↑	95% CI	GPU-s / trajectory ↓
Self-verifier	4B	58.90	$[57.60, 60.15]$	0.273
Self-verifier	8B	60.84	$[59.56, 62.14]$	0.299
Generative RM	4B	55.06	$[53.93, 56.18]$	2.693
Generative RM	8B	55.03	$[53.81, 56.25]$	2.930
Generative RM	32B	63.13	$[61.90, 64.36]$	3.115

E.2 Sensitivity to score-label cardinality

The expected-utility readout already produces continuous scores from three labels. To test whether a finer ordinal alphabet improves ranking, we compare A–C, A–E, and A–G using the same frozen Qwen3-4B verifier in FP32 on the Medical300 panel. Each alphabet assigns equally spaced utilities from 1 to 0 and prefills its midpoint label (B, C, or D, respectively). All variants use the same criterion weights and probability-weighted aggregation.

Table 9: **Sensitivity to score-label cardinality.** Prompt-macro averages over the 297 Medical300 prompts with non-constant reference scores. Spearman correlation uses average ranks for tied scores.

Score labels	Spearman ↑	Pair Acc. (%) ↑
A–C	**0.234**	**58.9**
A–E	0.138	55.2
A–G	0.135	55.4

The three-label interface achieves the highest agreement under both metrics (Table 9). Increasing the number of labels does not improve ordering on this panel, supporting the compact A/B/C readout used in the main method.