LEARNING FROM UNVERIFIED PEERS VIA SHARED RESIDUAL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model can generate several solutions to the same problem, some leading to correct answers and others to incorrect ones. Many post-training methods turn these trajectories into learning signals using supervision from reference answers, verifiable feedback, or external teachers. Without such supervision, learning from the model's own generations becomes less straightforward, as their correctness cannot be reliably assessed. An incorrect solution may nevertheless contain useful intermediate reasoning, so its value as supervision need not be determined solely by its final answer. Can a model improve by learning from its unverified peer solutions without first determining which are correct? We propose **Shared Residual Distillation (SRD)**, an on-policy self-distillation method that learns from the model's own unverified peer trajectories. SRD compares how peers change the prediction distribution relative to a control, retaining adjustments that agree in direction and are conservative in magnitude. This yields supervision without treating any complete solution as correct, and the trained model requires no peers at inference time. Starting from the corresponding base models, SRD raises mean AIME24–26 avg@12 from 11.02 to 16.20 on Qwen3-1.7B, from 19.07 to 45.46 on Qwen3-4B, and from 21.39 to 54.44 on Qwen3-8B. Controlled ablations support the value of the shared-residual construction, and we observe no degradation relative to the base models on two general-capability benchmarks. Our code is available at `https://anonymous.4open.science/r/SRD-E37E/`.

1 INTRODUCTION

Post-training has substantially improved the reasoning capabilities of language models. Reinforcement learning with verifiable rewards (RLVR) uses outcome feedback to form training signals; Group Relative Policy Optimization (GRPO), for example, learns from relative rewards within groups of sampled solutions (Shao et al., 2024). On-policy distillation (OPD) obtains dense token-level supervision from an external teacher along student-generated trajectories (Agarwal et al., 2024). On-policy self-distillation (OPSD) instead enables the model to provide its own supervision by conditioning its teacher on reference answers or solutions available during training (Zhao et al., 2026).

When these supervision sources are unavailable, the model's own generations remain an accessible resource. For the same problem, it can sample reasoning trajectories that follow different strategies and may reach different answers. Test-Time Reinforcement Learning (TTRL) uses majority-vote answers to construct rewards on unlabeled problems (Zuo et al., 2025). Majority voting thus provides a route to label-free post-training, but still establishes an answer-level reference against which to assign rewards.

A trajectory's final answer does not fully describe its value for learning. Even a solution that ends in an arithmetic or reasoning error can contain useful decompositions, relevant constraints, and valid intermediate derivations. Appendix B provides a concrete example of useful local guidance from peers with incorrect final answers. We investigate how to extract this local information when reference answers, verifiable feedback, and external teachers are unavailable:

Can a model learn from its own unverified peer solutions without first determining which answers are correct?

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.

We propose **Shared Residual Distillation (SRD)**, which turns agreement between peer-induced distributional changes into dense supervision for on-policy self-distillation (Figure 1). A frozen copy of the initial model evaluates student-generated prefixes while separately conditioned on uniformly sampled same-problem peers and a solution from another problem. The cross-problem solution supplies a common reference for comparing the peers' effects. SRD retains adjustments supported in the **same direction across peers**, with magnitude bounded by the **weakest supporting peer**. Applied to the frozen model's distribution under the student input, these adjustments form a teacher target that captures shared local guidance from the sampled solutions. The student learns from this target using only the problem and its own prefix, and requires no peers at inference time.

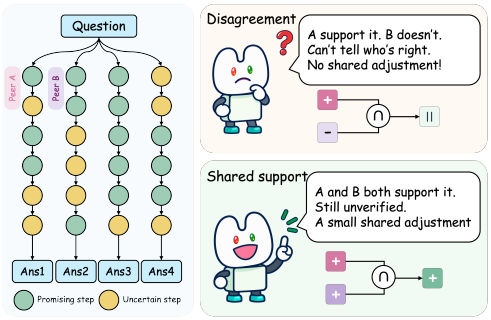

Figure 1: Intuition behind SRD. Sampled solutions may contain useful steps even when their answers are unverified. When peer effects disagree, SRD makes no shared adjustment; when they agree, it retains a conservative adjustment without deciding which peer is correct.

Empirically, SRD improves reasoning across model scales. Starting from the corresponding base models, SRD raises mean AIME24–26 avg@12 from 11.02 to 16.20 on Qwen3-1.7B, from 19.07 to 45.46 on Qwen3-4B, and from 21.39 to 54.44 on Qwen3-8B. The gains extend to the Harvard–MIT Mathematics Tournament benchmarks. Performance on Instruction-Following Eval (IFEval) and Massive Multitask Language Understanding Pro (MMLU-Pro) is maintained relative to the corresponding base models. Our contributions are summarized as follows:

- We show that unverified model-generated solutions can provide effective post-training supervision without reference answers, correctness rewards, or an external teacher.
- We introduce Shared Residual Distillation (SRD), which compares token predictions under each peer with those under a cross-problem control. It retains changes that agree in direction across peers and takes the smallest supported magnitude.
- We demonstrate mathematical reasoning gains across three Qwen3 model sizes while maintaining performance on the evaluated general-capability benchmarks. Ablations show the value of conservative magnitude selection and the cross-problem residual reference.

2 Related Work

Reward-based post-training and privileged distillation. Post-training learns from human preferences, verifiable rewards, or teacher predictions (Ouyang et al., 2022; Rafailov et al., 2023; Hinton et al., 2015). GRPO forms relative advantages from rewards within groups of sampled solutions (Shao et al., 2024); OPD instead uses dense teacher distributions along student-generated prefixes (Agarwal et al., 2024; Gu et al., 2024). Privileged information strengthens training-time supervision without changing test-time inputs (Vapnik & Vashist, 2009; Lopez-Paz et al., 2015), and OPSD applies this idea by conditioning a self-teacher on reference solutions (Zhao et al., 2026). CREDIT studies contrastive token credit under environment feedback (Shen et al., 2026). We retain the privileged distillation interface and study target construction when the extra information consists of unverified peer solutions.

Self-generated supervision for post-training. Self-Instruct expands instruction data through generation (Wang et al., 2023b), and self-consistency supplies high-confidence reasoning targets for self-improvement (Huang et al., 2023). Self-Rewarding Language Models use model-produced judgments for preference training (Yuan et al., 2024); TTRL converts answer consensus into rewards on unlabeled problems (Zuo et al., 2025). Answer consensus also guides self-distillation (Guo et al., 2026; Wang et al., 2026; Li et al., 2026). SRD samples unverified peers uniformly and constructs dense supervision from their shared residuals relative to a cross-problem context, without selecting or labeling trajectories by final-answer consensus. Table 1 compares supervision sources and granularity.

Table 1: Post-training supervision. Signal is token-level (dense) or outcome-level (sparse); Labels denotes answer-level supervision. –: no distillation teacher.

Method	Samples	Signal	Teacher	Labels
RLVR (GRPO)	✓ On-policy	✗ Sparse	–	Gold
External OPD	✓ On-policy	✓ Dense	✗ External	None
Reference OPSD	✓ On-policy	✓ Dense	✓ Self	Gold
TTRL	✓ On-policy	✗ Sparse	–	Pseudo
SRD (ours)	✓ On-policy	✓ Dense	✓ Self	**None**

Label-free inference-time improvement. Multiple samples can improve predictions without ground-truth answers at inference. Self-consistency aggregates candidate answers (Wang et al., 2022), while Tree of Thoughts supports search using model-evaluated intermediate states (Yao et al., 2023). These methods use additional computation to select or construct an answer. They motivate exploiting self-generated reasoning, but do not directly specify a training target. We use multiple solutions to construct local distributional supervision that is transferred into a student generating from the problem alone. Additional related work is provided in Appendix A.

3 Method

3.1 Preliminaries

Let $\mathcal{D}$ be a dataset of math problems and π_θ a language-model policy. For a problem $x \sim \mathcal{D}$, let $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$ denote a sampled solution. At position t, the context $h_t = (x, y_{<t})$ consists of the problem and previously generated tokens; $v \in \mathcal{V}$ denotes a candidate next token.

RLVR and GRPO. RLVR learns from a verifiable reward $R(x, y)$. GRPO (Shao et al., 2024) samples G trajectories for each problem from $\pi_{\theta_{\text{old}}}$. With $R_i = R(x, y^{(i)})$ and $h_t^{(i)} = (x, y_{<t}^{(i)})$, it computes a group-relative advantage and token-level probability ratio:

$$A_i = \frac{R_i - \text{mean}_j\, R_j}{\text{std}_j\, R_j + \epsilon_0}, \qquad \rho_{i,t} = \frac{\pi_\theta(y_t^{(i)} \mid h_t^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid h_t^{(i)})}, \tag{1}$$

where $\epsilon_0 > 0$ stabilizes the denominator. GRPO maximizes

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\left\{\min[\rho_{i,t}A_i,\ \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)A_i] - \beta\widehat{D}_{i,t}\right\}\right], \tag{2}$$

where ε controls ratio clipping and β weights an estimated Kullback–Leibler (KL) penalty $\widehat{D}_{i,t}$ against a fixed reference policy. Group normalization avoids a learned value baseline, but still requires rewards to define A_i.

OPD and privileged self-distillation. OPD (Agarwal et al., 2024) trains the student using teacher predictions along its own sampled trajectories. Let $\mathcal{B}$ contain trajectories sampled from the current policy $\pi_{\theta_{\text{old}}}$, with T_i scored positions and prefixes $h_t^{(i)} = (x^{(i)}, y_{<t}^{(i)})$. A standard formulation uses the forward KL divergence with a token-mean objective:

$$\mathcal{L}_{\text{OPD}}(\theta; q) = \mathbb{E}\left[\frac{1}{\sum_{i\in\mathcal{B}} T_i}\sum_{i\in\mathcal{B}}\sum_{t=1}^{T_i}\text{KL}\Big(\text{sg}[q(\cdot \mid h_t^{(i)})]\ \Big\|\ \pi_\theta(\cdot \mid h_t^{(i)})\Big)\right], \tag{3}$$

where sg denotes stop-gradient. Trajectories and teacher targets remain fixed during the update.

OPSD (Zhao et al., 2026) conditions a frozen self-teacher $\pi_{\bar{\theta}}$ on a training-only context z, such as a reference solution, while leaving the student's input unchanged:

$$q_z(v \mid h_t) = \pi_{\bar{\theta}}(v \mid x, z, y_{<t}). \tag{4}$$

This follows the privileged-information setting (Vapnik & Vashist, 2009; Lopez-Paz et al., 2015).

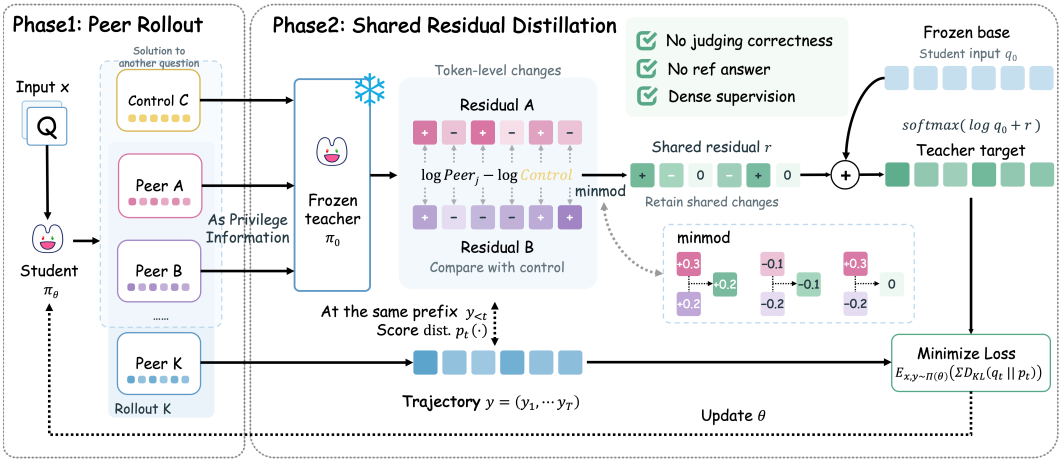

Figure 2: Overview of SRD. **Phase 1: Peer rollout.** The student samples solutions for each problem, split into peer contexts and target trajectories. **Phase 2: Shared residual distillation.** A frozen teacher compares peer-conditioned predictions with a cross-problem control and retains conservative shared adjustments. These adjustments modify the frozen model's question-only base distribution to form targets for student updates.

3.2 Shared Residual Distillation

Motivation. Presenting a solution as context can affect both problem-relevant predictions and generic response patterns. We use a cross-problem solution as a common reference and compare how each same-problem peer changes the teacher's predictions relative to that reference. Agreement determines which residual coordinates are retained, and the weakest supporting peer bounds their magnitude.

Peer contexts. For each problem $x \sim \mathcal{D}$, sample $K > M$ independent trajectories $Y = \{y^{(1)}, \ldots, y^{(K)}\}$ from the current policy. Choose a peer-index set $\mathcal{P} \subset \{1, \ldots, K\}$ uniformly among all subsets of size M, and use the remaining indices $\mathcal{T} = \{1, \ldots, K\} \setminus \mathcal{P}$ as distillation targets. Thus, each group contributes M peers and $K - M$ held-out target trajectories, and a target trajectory is never its own privileged context.

For the control, randomly sample a trajectory z_c generated for another problem $x_c \neq x$ in the same batch. It reuses an existing rollout and serves as the common reference for all M peer residuals, rather than an additional same-problem peer.

At position t of a target trajectory $i \in \mathcal{T}$, write $h = (x, y^{(i)}_{<t})$ for its problem and generated prefix. Let $s(x)$ be the student's standard problem prompt and $c(x, z)$ the teacher prompt containing a solution z and guidance on using it. With $z_j = y^{(j)}$, the same frozen initial model gives $M + 2$ next-token distributions:

$$\begin{aligned} q_0(v \mid h) &= \pi_{\bar{\theta}}(v \mid s(x), y^{(i)}_{<t}), \\ q_c(v \mid h) &= \pi_{\bar{\theta}}(v \mid c(x, z_c), y^{(i)}_{<t}), \\ q_j(v \mid h) &= \pi_{\bar{\theta}}(v \mid c(x, z_j), y^{(i)}_{<t}), \qquad j \in \mathcal{P}. \end{aligned} \tag{5}$$

The base q_0 uses exactly the student's question-only prompt and target prefix, without additional teacher-side instructions, and is evaluated with frozen initial parameters $\bar{\theta}$. The current student distribution uses the same input with trainable parameters θ. Each q_j uses a same-problem peer, whereas q_c uses the cross-problem trajectory in the same teacher prompt. All distributions score the current problem x and the same target prefix, without its future continuation; Appendix C.4 gives the prompts and control substitution.

Shared residual distillation. Fix h throughout the following construction and suppress it in the notation. For each same-problem peer, we measure the log-probability change relative to the

cross-problem control q_c and subtract its q_0-weighted average across the vocabulary:

$$d_j(v) = \log \frac{q_j(v)}{q_c(v)}, \qquad \delta_j(v) = d_j(v) - \sum_{u \in \mathcal{V}} q_0(u) d_j(u), \qquad j \in \mathcal{P}. \tag{6}$$

Thus q_c is the residual reference, while q_0 supplies the centering weights and the final target's base. Each $\delta_j(v)$ is a scalar centered change for one candidate token; its sign indicates whether the peer-versus-control change lies above or below its q_0-weighted average. We compare the residuals *token by token*: when all M are positive or all are negative, retain the smallest magnitude; otherwise retain zero:

$$r_M(v) = \begin{cases} \min\limits_{j \in \mathcal{P}} \delta_j(v), & \min\limits_{j \in \mathcal{P}} \delta_j(v) > 0, \\ \max\limits_{j \in \mathcal{P}} \delta_j(v), & \max\limits_{j \in \mathcal{P}} \delta_j(v) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

For $M = 2$, this is the minmod rule: same sign, smaller magnitude; otherwise zero. At each vocabulary coordinate, the residual magnitude is bounded by the weakest supporting peer-versus-control residual. We apply it to q_0, the frozen model's distribution under the student input, to form the teacher target:

$$q_\cap(v) = \frac{q_0(v) \exp(r_M(v))}{\sum_{u \in \mathcal{V}} q_0(u) \exp(r_M(u))}. \tag{8}$$

Repeating this construction at each target prefix provides teacher targets for the question-only student.

Training objective. We inherit OPSD's pointwise-clipped distillation objective (Zhao et al., 2026), replacing its reference-conditioned teacher with $q_\cap$. For teacher and student distributions q and p, define

$$\ell_\tau(q, p) = \sum_{v \in \mathcal{V}} \min\left\{ \bar{q}(v) \log \frac{\bar{q}(v)}{p(v)}, \tau \right\}, \qquad \bar{q} = \mathrm{sg}[q]. \tag{9}$$

The cap τ applies to each vocabulary-coordinate contribution before summation; removing it recovers forward KL. Over the batch $\mathcal{B}$ of held-out target trajectories, we minimize

$$\mathcal{L}_{\mathrm{SRD}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{i \in \mathcal{B}} T_i} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T_i} \ell_\tau\Big(q_\cap(\cdot \mid h_t^{(i)}), \pi_\theta(\cdot \mid h_t^{(i)}) \Big) \right]. \tag{10}$$

Here T_i counts valid target tokens, and the expectation covers rollout, peer, and cross-problem control sampling.

Teacher targets are detached, and the student receives only the problem and its own prefix. Peers, controls, trajectories, and targets remain fixed within an update; fresh groups are sampled from the updated policy. We retain OPSD's clipping threshold and training settings, listed in Appendix C.2. Algorithm 1 summarizes the workflow.

The shared target captures agreement relative to a common control at the *current target prefix*, without selecting a complete peer solution as ground truth. At inference, the trained student generates from the problem alone, with no peer sampling or extra teacher calls.

4 Experiment

Our experiments examine whether shared residuals from unverified peers improve mathematical reasoning, how this supervision compares with reference-based and label-free post-training, and whether the resulting models retain capabilities beyond mathematics.

4.1 Experiment Setup

We evaluate the Qwen3 family at three model sizes: 1.7B, 4B, and 8B parameters. All main-table SRD results use $K = 4$ trajectories per problem and $M = 2$ peers, leaving two trajectories as distillation targets. A trajectory from another problem in the batch provides the residual control without an additional rollout; Appendix D.3 varies K, rollout length, and M.

Algorithm 1 Shared Residual Distillation

Require: Student π_θ; frozen model $\pi_{\bar{\theta}}$; problems $\mathcal{D}$; $K > M \geq 2$.
Notation: $\mathcal{P}_x$: M peers for x; $\mathcal{B}$: held-out targets; $\mathcal{H}(\mathcal{B})$: scored target prefixes.
$\mathrm{Center}_{q_0}(d) = d - \sum_v q_0(v)d(v)$; ℓ_τ: clipped loss in Eq. (9).
for each iteration **do**
// Step 1: sample peer rollouts
$\mathcal{X} \sim \mathcal{D}, \quad |\mathcal{X}| \geq 2; \qquad Y_x \leftarrow \mathrm{Rollout}(\pi_\theta, x, K), \quad \forall x \in \mathcal{X}$
$(\{\mathcal{P}_x\}, \mathcal{B}) \leftarrow \mathrm{UniformSplit}(\{Y_x\}, M)$
// Step 2: construct shared targets
for each target $(x, y) \in \mathcal{B}$ **do**
$z_c \sim \mathrm{Uniform}\Big(\bigcup_{x' \neq x} Y_{x'}\Big)$
for each scored prefix h of (x, y) **do**
$(q_0, q_c, \{q_j\}) \leftarrow \mathrm{Score}_{\bar{\theta}}(h, \mathcal{P}_x, z_c)$ Eq. (5)
$\delta_j \leftarrow \mathrm{Center}_{q_0}(\log q_j - \log q_c), \quad j \in \mathcal{P}_x$
$r \leftarrow \mathrm{minmod}(\{\delta_j\}_{j \in \mathcal{P}_x})$
$q_\cap(\cdot \mid h) \leftarrow \mathrm{sg}[\mathrm{softmax}(\log q_0 + r)]$
end for
end for
// Step 3: update the student
$\mathcal{L} \leftarrow \mathrm{Mean}_{h \in \mathcal{H}(\mathcal{B})}\, \ell_\tau(q_\cap(\cdot \mid h), \pi_\theta(\cdot \mid h))$
$\theta \leftarrow \mathrm{OptimizerStep}(\theta, \nabla_\theta \mathcal{L})$
end for

Datasets and evaluation benchmarks. We use the OpenThoughts mathematics subset released with OPSD (Zhao et al., 2026), containing approximately 29.4K problem–solution pairs. SRD uses only the problems; reference solutions are available to the reference-based baselines. We evaluate in-domain (ID) mathematical reasoning on AIME24–26, HMMT25/26, and MATH-500 (Lightman et al., 2024), and out-of-domain (OOD) retention on IFEval and MMLU-Pro. The OOD benchmarks assess instruction following and broader knowledge-intensive reasoning, respectively. We checked the training set against all evaluation benchmarks and found no overlapping problems. All training and evaluation use non-thinking mode. We report avg@12 on AIME and HMMT, pass@1 on MATH-500, strict prompt-level accuracy on IFEval, and overall accuracy on MMLU-Pro. More details are provided in Appendix C.1.

Baselines. We compare with GRPO (Shao et al., 2024) with answer-verified outcome rewards, OPSD (Zhao et al., 2026) with reference solutions as privileged teacher context, and TTRL (Zuo et al., 2025) with majority-vote pseudo-rewards. GRPO and TTRL share the same 8192-token rollout and optimization configuration, differing only in whether rewards use verified answers or majority-derived pseudo-labels. OPSD‡ samples two independent target trajectories per problem so that its target batch matches SRD; Appendix D.5 separately measures the change from the original one-trajectory configuration across all three model sizes. Full settings are provided in Appendix C.

4.2 Main Results

Table 2 organizes the comparison across three model sizes, mathematical reasoning, and out-of-domain capabilities. The reference-based baselines contextualize performance with access to reference solutions, while TTRL provides a label-free comparison based on inferred answer-level rewards. For all model sizes, we evaluate checkpoints fixed in advance: step 50 for OPSD and SRD, and step 300 for GRPO and TTRL. Each method uses the same checkpoint across all benchmarks.

Table 2: Main results across three Qwen3 model sizes (%; higher is better). AIME and HMMT report avg@12; MATH-500 reports pass@1. Bold marks the best result per model and benchmark. †: TTRL trained on the common training split; ‡: OPSD with two target trajectories per problem.

	Mathematics (ID)						General (OOD)	
Method	**AIME24**	**AIME25**	**AIME26**	**HMMT25**	**HMMT26**	**MATH-500**	**IFEval**	**MMLU-Pro**
Qwen3-1.7B								
Base	13.33	8.61	11.11	5.97	8.08	74.00	66.91	43.87
GRPO	15.28	10.83	8.89	5.69	10.86	**75.80**	67.84	44.68
TTRL†	14.44	11.67	10.28	3.89	8.59	73.80	**68.58**	44.51
OPSD‡	14.17	11.39	10.56	5.97	8.33	72.60	67.10	44.17
SRD (ours)	**21.11**	**15.00**	**12.50**	**6.11**	**17.17**	74.80	67.10	**45.31**
Qwen3-4B								
Base	21.11	20.83	15.28	10.56	15.91	84.20	79.30	60.61
GRPO	26.39	20.00	19.17	12.08	18.43	85.80	78.37	61.17
TTRL†	24.17	19.17	18.33	10.28	15.66	83.80	78.19	60.74
OPSD‡	22.50	26.39	25.00	10.14	18.43	83.60	79.67	60.05
SRD (ours)	**47.78**	**41.11**	**47.50**	**25.14**	**23.48**	**87.80**	**80.22**	**61.52**
Qwen3-8B								
Base	26.94	21.39	15.83	11.53	20.71	83.20	81.70	64.98
GRPO	27.50	23.89	17.78	12.64	18.94	85.60	80.78	65.37
TTRL†	24.17	25.83	17.22	12.64	17.68	87.20	80.78	65.86
OPSD‡	40.56	31.94	27.50	18.06	24.24	88.20	82.26	65.98
SRD (ours)	**59.72**	**51.94**	**51.67**	**35.97**	**35.10**	**93.60**	**82.81**	**66.25**

Starting from the corresponding base models, SRD raises mean AIME24–26 avg@12 from 11.02 to 16.20 on Qwen3-1.7B, from 19.07 to 45.46 on Qwen3-4B, and from 21.39 to 54.44 on Qwen3-8B. The improvements also extend to HMMT, with no observed degradation relative to the corresponding base models on the two out-of-domain benchmarks. Appendix C.5 additionally reports pass@12 results on AIME and HMMT.

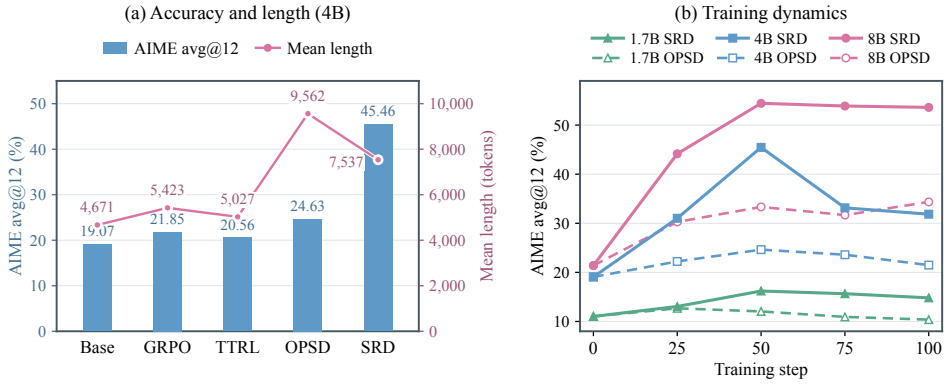

Figure 3: AIME accuracy, response length, and training dynamics. (a) Mean AIME24–26 avg@12 (bars) and response length (line) on Qwen3-4B. (b) Mean AIME24–26 avg@12 throughout training for OPSD (dashed) and SRD (solid) at three model sizes.

Figure 3(a) compares AIME accuracy and mean response length for the five Qwen3-4B methods at the main-table checkpoints. Mean AIME24–26 avg@12 rises from 19.07 for Base to 45.46 with SRD, while mean response length increases from 4,671 to 7,537 tokens. OPSD reaches 24.63 avg@12 with 9,562 tokens on average. SRD achieves higher AIME accuracy with shorter responses than OPSD.

Table 3: Qwen3-4B target-construction ablations with clipping at the main-table checkpoint (avg@12, %). Average is the mean over AIME24–26.

Variant	AIME24	AIME25	AIME26	Average
Target construction				
Mean residual	36.67	29.17	36.11	33.98
Random residual mask (matched sparsity)	41.67	32.50	41.67	38.61
Signed-mask mean	38.89	32.78	40.83	37.50
Shared residual (q_0 reference)	40.28	40.56	42.22	41.02
Full SRD (ours)	47.78	41.11	47.50	45.46

Panel (b) tracks both methods throughout training: SRD remains above OPSD at every evaluated post-initialization step across all three model sizes. Both methods show some decline after their initial gains. On Qwen3-4B, mean avg@12 falls from 45.46 to 31.85 for SRD and from 24.63 to 21.48 for OPSD between steps 50 and 100. On Qwen3-1.7B, OPSD reaches 10.37 at step 100, below the base model's 11.02. SRD on Qwen3-8B retains most of its gains, with mean avg@12 moving only slightly from 54.44 at step 50 to 53.61 at step 100. These patterns indicate sensitivity to model size and training duration in OPSD-based distillation. The stronger retention at 8B suggests that larger models may interpret privileged context more effectively and sustain more stable learning.

4.3 Ablation Studies

Our central claim is that useful supervision can be extracted from the *shared effects relative to a cross-problem control* of unverified peers, without selecting a preferred solution. We first isolate the target construction on Qwen3-4B, holding the training data, student initialization, rollout budget, target count, peer sampling, and evaluation protocol fixed (Table 3). We then examine sensitivity to the inherited clipping objective across all three model sizes.

Target construction. We first examine how shared direction and conservative magnitude contribute to the teacher target. The mean-residual variant scores the same four teacher distributions as full SRD but replaces the intersection in Eq. (7) with $(\delta_a + \delta_b)/2$. A matched-sparsity control randomly masks this mean residual to match the nonzero support size of SRD, testing whether undirected sparsification is sufficient. Signed-mask mean instead uses the same agreement test as SRD to zero sign-disagreeing residual coordinates, but averages the surviving same-sign residuals rather than bounding them by the weakest supporting residual. It isolates the value of conservative magnitude selection under the same sign mask.

Full SRD reaches 45.46 mean AIME24–26 avg@12, compared with 37.50 for signed-mask averaging and 38.61 for matched-sparsity random masking. The comparison with signed-mask averaging holds the agreement rule fixed and isolates the contribution of conservative magnitude selection. These results support combining shared direction with a magnitude bounded by the weakest supporting residual.

Residual reference. Replacing q_c with q_0 in Eq. (6), while retaining the same centering, shared-residual rule, and q_0-based target, yields 41.02 mean AIME24–26 avg@12, compared with 19.07 for the base model and 45.46 for full SRD. Shared peer residuals provide substantial improvement with either reference, with the cross-problem control yielding the stronger result.

Effect of pointwise clipping. OPSD introduced pointwise divergence clipping to stabilize distillation (Zhao et al., 2026), and SRD inherits this mechanism. Table 4 removes clipping from both methods while retaining their respective targets and training settings, with evaluation at the main-table checkpoint. Removing clipping lowers mean AIME24–26 avg@12 for both methods at every model size, consistent with the degradation reported in OPSD's clipping ablation. Both methods therefore achieve their strongest results by combining their respective teacher targets with clipping.

Under matched clipping conditions, SRD has higher mean AIME24–26 avg@12 than OPSD at every model size, both with and without clipping. Even without clipping, SRD consistently exceeds both

Table 4: OPSD and SRD without clipping (%). Average is the mean over AIME24–26; bold marks column-wise best results within each model.

Method	AIME24		AIME25		AIME26		Average	
	avg@12	pass@12	avg@12	pass@12	avg@12	pass@12	avg@12	pass@12
Qwen3-1.7B								
OPSD‡	11.39	30.00	11.11	23.33	9.17	**23.33**	10.56	25.55
SRD (ours)	**12.78**	**43.33**	**11.67**	**30.00**	**11.67**	20.00	**12.04**	**31.11**
Qwen3-4B								
OPSD‡	**23.61**	43.33	19.72	30.00	16.11	36.67	19.81	36.67
SRD (ours)	**23.61**	**50.00**	**22.22**	**56.67**	**20.83**	**50.00**	**22.22**	**52.22**
Qwen3-8B								
OPSD‡	29.17	53.33	20.00	33.33	18.06	**53.33**	22.41	46.66
SRD (ours)	**30.83**	**56.67**	**23.61**	**53.33**	**19.72**	**53.33**	**24.72**	**54.44**

OPSD and the base model on this metric across all three sizes: mean avg@12 rises from the base models' scores of 11.02, 19.07, and 21.39 to 12.04, 22.22, and 24.72 for Qwen3-1.7B, 4B, and 8B, respectively. Against unclipped OPSD, it wins eight of nine individual avg@12 comparisons and ties the remaining one; for pass@12, it wins seven, ties one, and loses one. On AIME25, pass@12 rises from 30.00% to 56.67% for Qwen3-4B and from 33.33% to 53.33% for Qwen3-8B. These comparisons show that the benefit of shared-residual targets persists when the inherited clipping mechanism is removed.

On Qwen3-4B, the q_0-reference variant without clipping reaches 18.61 mean avg@12, slightly below the base model's 19.07. With q_c, the score rises from the base model's 19.07 to 22.22. This comparison is consistent with a stabilizing contribution from the cross-problem control when clipping is absent. Appendix D.3 reports sampling-budget and peer-count sensitivity, including Figure 4.

5 Conclusion

We study how a language model can learn from its own unverified solutions without reference answers, correctness labels, or selecting a winning trajectory. We introduce Shared Residual Distillation, which uses independent peer solutions as privileged teacher contexts and constructs local supervision from their shared, conservatively bounded residuals relative to a randomly sampled cross-problem control, applied to the frozen model's distribution under the student input. Starting from the corresponding base models, SRD raises mean AIME24–26 avg@12 from 11.02 to 16.20 on Qwen3-1.7B, from 19.07 to 45.46 on Qwen3-4B, and from 21.39 to 54.44 on Qwen3-8B. The improvements extend to HMMT, with no observed degradation relative to the base models on the evaluated out-of-domain benchmarks. These results show that unverified peer generations can provide effective dense post-training supervision even when no trajectory is promoted as a pseudo-label.

6 Limitations

Peer agreement does not guarantee correct reasoning; SRD uses it to identify shared local adjustments without assigning correctness labels to entire solutions. The cross-problem control is intended to reduce generic distributional shifts introduced by additional solution text and make the supervision more specific to the current problem. The extent to which it removes stylistic effects remains unclear, and the control's own content may introduce effects that alter residual directions. Our empirical study focuses on mathematical reasoning, and the effectiveness of SRD on open-ended tasks remains untested.

Ethics Statement

This work studies language-model post-training using publicly available datasets, benchmarks, and pretrained models. It involves no human-subject experiments or collection of personal data. Improved reasoning capabilities may support education and scientific work, but language models can also produce misleading or incorrect content. Downstream applications should use verification and safeguards appropriate to their intended use.

Reproducibility Statement

Section 3 defines the target construction and training objective, and Algorithm 1 summarizes the training workflow. Section 4.1 and Appendix C.1 document the datasets, benchmarks, decoding settings, and evaluation metrics. Training and baseline configurations are given in Tables 5 and 6. Appendix C.4 provides the exact prompts, and Appendix D describes the ablation protocols. Our code is available at `https://anonymous.4open.science/r/SRD-E37E/`.

AI Use Statement

We used large language models to polish the manuscript and improve the quality of the writing. The peer trajectories and teacher predictions used by SRD are generated by the Qwen3 models as part of the training procedure described in Section 3. The authors reviewed and revised the AI-assisted materials and take responsibility for the final content of this work.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born Again Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1607–1616, 2018. URL `https://proceedings.mlr.press/v80/furlanello18a.html`.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/fef126561bbf9d4467dbb8d27334b8fe-Paper-Conference.pdf`.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Jiaxin Guo, Yanwei Yue, Xuanbo Fan, Chunyu Yang, and Yan Zhang. Learning from consensus and disagreement: Unsupervised on-policy self-distillation with minority-trajectory contrast. *arXiv preprint arXiv:2608.08764*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. URL https://arxiv.org/abs/2305.02301.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068, Singapore, December 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.emnlp-main.67. URL https://aclanthology.org/2023.emnlp-main.67/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. doi:10.1162/tacl_a_00713. URL https://aclanthology.org/2024.tacl-1.78/.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 54523–54549, 2025.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision. *arXiv preprint arXiv:2608.06296*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation. *arXiv preprint arXiv:2605.11613*, 2026.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi:10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Aozhe Wang, Zhengxi Lu, Jianze Wang, Shangke Lv, Ying Liu, Weiming Lu, Jun Xiao, Yueting Zhuang, Hua Yang, Qianglong Chen, et al. Ttpo: Test-time policy optimization. *arXiv preprint arXiv:2608.27448*, 2026.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5546–5558, 2023a.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi:10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

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

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. Self-contrast: Better reflection through inconsistent solving perspectives. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3602–3622, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.197. URL https://aclanthology.org/2024.acl-long.197/.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328. IEEE, 2018.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, yuchen zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 131459–131483. Curran Associates, Inc., 2025. doi:10.52202/085713-4376. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/be690ea16f005c174f6c4102a5970e67-Paper-Conference.pdf.

A Additional Related Work

Reasoning, verification, and feedback. Chain-of-thought prompting exposes intermediate computation (Wei et al., 2022; Kojima et al., 2022), while outcome rewards and process supervision turn evaluations of reasoning into training signals (Shao et al., 2024; Lightman et al., 2024; Wang et al., 2024a). The Self-Taught Reasoner (STaR) bootstraps from generated rationales filtered using answer correctness (Zelikman et al., 2022); this differs from learning without correctness labels. Feedback can also be expressed in language: Self-Refine iterates self-feedback (Madaan et al., 2023), Reflexion uses verbal feedback across attempts (Shinn et al., 2023), and tool-interactive critiquing incorporates external checks (Gou et al., 2024). Multiagent debate and Self-Contrast use interaction or contrast among solutions to guide subsequent responses (Du et al., 2023; Zhang et al., 2024). Their effectiveness depends on the model, task, and feedback source (Huang et al., 2024; Kamoi et al., 2024); reinforcement learning can explicitly train correction behavior (Kumar et al., 2025). These are broader ways to exploit generated reasoning, rather than components of our residual target construction.

Further distillation background. Sequence-level distillation transfers generated sequences (Kim & Rush, 2016), while Born Again Networks and Deep Mutual Learning demonstrate useful transfer between similarly parameterized models (Furlanello et al., 2018; Zhang et al., 2018). On-policy distillation follows the broader motivation of learning on the learner's state distribution (Ross et al., 2011). Teacher-generated rationales support reasoning transfer (Hsieh et al., 2023; Ho et al., 2023; Wang et al., 2023a). Weak-to-strong generalization further shows that imperfect supervision need not impose a hard ceiling on a pretrained student's performance (Burns et al., 2023). These results motivate learning from fallible outputs; our focus is how multiple such outputs should influence a single distributional target.

CoDA uses a representative answer-consensus trajectory as privileged context and adds a reference-anchored, KTO-style penalty for minority trajectories (Guo et al., 2026). TTPO conditions on a majority-vote pseudo-answer, distills agreeing rollouts, and penalizes disagreeing rollouts through grouped RL, with token weighting and masking in the respective branches (Wang et al., 2026). u-OPSD accepts pseudo-solutions using a self-consistency threshold and distills the resulting teacher on disagreeing trajectories (Li et al., 2026). These approaches use final-answer consensus to construct or route supervision. SRD instead samples peers uniformly and checks agreement between their centered residuals at each target prefix and vocabulary coordinate; peers can support the same local adjustment even when their final answers differ.

Consistency and distributional differences. Mean Teacher and FixMatch exploit consistency across views, together with labeled supervision, to learn from unlabeled examples (Tarvainen & Valpola, 2017; Sohn et al., 2020); deep ensembles aggregate predictions from multiple models (Lakshminarayanan et al., 2017). Our peers are unverified solutions used as contexts for one frozen model, and we combine their *changes relative to a common cross-problem control*, then apply the shared adjustment to the frozen model's distribution under the student input. Differences between predictive distributions also underlie contrastive decoding, context-aware decoding, and Decoding by Contrasting Layers (DoLa) (Li et al., 2023; Shi et al., 2024; Chuang et al., 2024). Those methods steer inference using model, context, or layer contrasts. Related interference-reduction ideas appear in gradient surgery and sign-aware model merging (Yu et al., 2020; Yadav et al., 2023). SRD applies a coordinatewise agreement rule to peer-induced log-probability residuals at student prefixes and uses the resulting teacher target for on-policy distillation. Its specific object is a shared conditional shift, rather than an averaged prediction, a merged parameter vector, or an inference-time decoding rule.

CREDIT studies self-distillation conditioned on environment feedback and subtracts a batch-estimated generic teacher component to refine token-level credit (Shen et al., 2026). Its contrast changes the *problem input* while holding the response prefix and feedback fixed, and the resulting signal enters a reverse-KL update. SRD holds the current problem and prefix fixed and changes the *solution context*, comparing each same-problem peer with an unrelated trajectory. It then retains shared centered residuals through minmod and normalizes the adjusted base q_0 into a distillation target. Thus our construction combines a context contrast with agreement across peers, using generated solutions without environment feedback.

B Case Study: Local Guidance from Incorrect Peers

Peers A/B answer incorrectly (correct: 429); C solves another problem. Ellipses mark omissions.

Question

Adam and Bettie are playing a game. They take turns generating a random number between 0 and 127 inclusive. The numbers they generate are scored as follows:

- If the number is zero, it receives no points.
- If the number is odd, it receives one more point than the number one less than it.
- If the number is even, it receives the same score as the number with half its value.

If Adam and Bettie both generate one number, the probability that they receive the same score is $\frac{p}{q}$ for relatively prime positive integers p and q. Find p.

Peer A

...Numbers from 0 to 127 (inclusive) are all 7-bit numbers: 0000000 to 1111111 in binary.

...

Sum of squares:

$$1 + 49 + 441 + 1225 + 1225 + 441 + 49 + 1 = \mathbf{3912}$$

Then:

$$P = \frac{3912}{128^2} = \frac{3912}{16384}$$

...

$$\frac{3912}{16384} = \frac{489}{2048}$$

Final answer: $\boxed{489}$

Peer B

...Note that 127 is $2^7 - 1 = 1111111_2$, so numbers from 0 to 127 include all 7-bit binary numbers (with leading zeros allowed), ...

$$\sum_{k=0}^{7} \binom{7}{k}^2 = \binom{14}{7}$$

Compute:

$$\binom{14}{7} = \cdots = \mathbf{12376}$$

...

$$\frac{12376}{16384} = \frac{1547}{2048}$$

Final answer: $\boxed{1547}$

Control C (cross-problem)

We are given the equation:

$$\frac{252}{\Theta} = \underline{3\Theta} + \Theta$$

...

$$\underline{3\Theta} = 10 \cdot 3 + \Theta = 30 + \Theta$$

...

$$\Theta^2 + 15\Theta - 126 = 0$$

...

$$\Theta = \frac{-15 \pm 27}{2}$$

...

But Θ is a digit, so it must be between 0 and 9.

Final answer: $\boxed{6}$

Student (excerpt)

We are given a game between Adam and Betty. ... So the score of a number n is the number of 1s in its binary representation. ... Each number from 0 to 127 has **8** bits (since $2^8 = 256$), so we can list how many numbers have a certain number of 1s in their 8-bit binary representation.

Next-token probabilities at the highlighted student position.

Privileged context / target	$P(7)$ (%)	$P(8)$ (%)
Student input (q_0)	18.13	79.45
Cross-problem control (q_c)	13.54	83.06
Peer A (q_A)	99.96	0.01
Peer B (q_B)	99.90	0.04
SRD ($q_\cap$)	99.83	0.06
Reference solution (q_{ref})	99.94	0.01

The frozen model uses the student's question-only prompt for q_0 and the cross-problem solution C as auxiliary context for q_c. All rows score the prefix ending in "Each number from 0 to 127 has", using full-vocabulary softmax at temperature 1.1. The control does not reverse the token preference: both q_0 and q_c favor the incorrect 8 over 7. Both peers retain the correct seven-bit representation despite later arithmetic errors; their shared-residual target assigns 99.83% to 7. In a separate greedy continuation from this prefix, forcing only 7 and then continuing without peers yields the correct answer 429, versus 6435 without the intervention. This example illustrates that final-answer correctness is not a prerequisite for useful local guidance. Peers with incorrect answers can still provide shared local signals that survive subtraction of the cross-problem control and conservative residual aggregation.

C Experimental Details

C.1 Data and Evaluation Protocol

Training data and supervision. We use the released OPSD subset, `siyanzhao/Openthoughts_math_30k_opsd`.[1] We use its mathematical problem statements as the common source of training inputs. OPSD supplies the released `solution` field only to its teacher, and GRPO verifies generated answers against the reference. Neither SRD nor the TTRL adaptation uses reference solutions, correctness annotations, or benchmark labels for optimization. Benchmark problems are reserved for evaluation.

Benchmark versions. The American Invitational Mathematics Examination (AIME) benchmark combines the first and second examinations of each year, giving 30 problems per annual benchmark. For HMMT25, we evaluate both the February 2025 and November 2025 Harvard–MIT Mathematics Tournament (HMMT) releases and report the arithmetic mean of their two benchmark scores. HMMT26 uses the February 2026 release. We use the MathArena releases for HMMT25/26 and AIME26;[2] the HMMT26 set contains 33 entries and is evaluated in full. MATH-500 uses the 500-problem test subset distributed by HuggingFaceH4.[3] Instruction-Following Eval (IFEval; Zhou et al., 2023) uses the original released prompts and deterministic instruction checkers, while Massive Multitask Language Understanding Pro (MMLU-Pro; Wang et al., 2024b) uses its full test split and five subject-matched demonstrations from its validation split.[4] ID and OOD are defined relative to the mathematical adaptation domain; IFEval and MMLU-Pro assess general capabilities after mathematical post-training.

Generation and metrics. Native chat templates are used with thinking disabled for student sampling, teacher scoring, and evaluation. Mathematical evaluations are zero-shot, with a request to place the final answer in `\boxed{}`. We use a generation budget of 32,768 tokens, a context limit of 40,960, temperature 0.7, top-p 0.8, top-k 20, and presence penalty 1.0. Evaluation runs with vLLM 0.11, one complete model per device, and tensor parallelism of one; independent problem shards provide data parallelism. All compared methods use the same evaluator, prompts, and per-benchmark decoding settings.

For N problems, let c_{ij} indicate whether response j to problem i is correct. The reported sampling accuracy is

$$\text{avg@12} = \frac{100}{12N} \sum_{i=1}^{N} \sum_{j=1}^{12} c_{ij}. \tag{11}$$

This is an estimate of single-sample accuracy, not pass@12 or majority-vote accuracy. MATH-500 uses one response per problem under the same mathematical decoding settings, yielding pass@1. Mathematical correctness is evaluated from the last boxed answer with symbolic equivalence checking through Math-Verify.[5] Missing or unparseable answers count as incorrect. IFEval and MMLU-Pro use one greedily decoded response, with an 8,192-token output budget and no added math-specific instruction. IFEval's primary score requires every verifiable instruction in a prompt to pass the strict checker; MMLU-Pro reports the fraction of test questions with the correct selected option.

C.2 Training Configurations

Shared distillation setup. Within each model size, OPSD[‡] and SRD use a frozen copy of that model's initialization as teacher and keep the student's prompt question-only. The Qwen3-4B configuration uses Low-Rank Adaptation (LoRA; Hu et al., 2021) with rank 64 and scaling factor 128 on the query, key, value, output, gate, up, and down projections. Both methods use a learning rate of 5×10^{-6}, a linear schedule over 100 optimizer steps, and maximum gradient norm 0.1. Both

[1] `https://huggingface.co/datasets/siyanzhao/Openthoughts_math_30k_opsd`

[2] `https://huggingface.co/MathArena`

[3] `https://huggingface.co/datasets/HuggingFaceH4/MATH-500`

[4] `https://github.com/google-research/google-research/tree/master/instruction_following_eval`; `https://github.com/TIGER-AI-Lab/MMLU-Pro`

[5] `https://github.com/huggingface/Math-Verify`

use the pointwise-clipped objective defined in Eqs. (9)–(10), following OPSD (Zhao et al., 2026). Student and teacher distributions use scoring temperature 1.1, and the coordinatewise upper cap is $\tau = 10^{-6}$(1.7B and 4B, while 8B is $\tau = 10^{-7}$). Full-vocabulary summation precedes averaging over valid scored tokens. OPSD uses its reference-conditioned teacher target in place of $q_{\cap}$. Table 5 separates sampling budgets from scored prefixes.

Table 5: Training configuration for Qwen3-4B distillation. OPSD‡ matches SRD's number of scored target trajectories without generating unused peer contexts. Both methods use a 100-step schedule.

Setting	**OPSD‡**	**SRD**
Problems per update	32	32
Rollouts per problem	2	4
Target trajectories per update	64	64
Maximum rollout length	1,024	1,024
Maximum scored target prefix	1,024	1,024
Maximum training context	20,000	20,000
Rollout temperature / top-p / top-k	1.1 / 0.95 / 20	1.1 / 0.95 / 20
Frozen teacher contexts	Reference solution	None / peers a, b / control c
Learning rate	5×10^{-6}	5×10^{-6}
LoRA rank / scaling	64 / 128	64 / 128
Coordinate clipping threshold	$10^{-6}/10^{-7}$	$10^{-6}/10^{-7}$
Maximum gradient norm	0.1	0.1

Rationale for inherited clipping. Zhao et al. (2026) report that a small set of stylistic tokens can contribute disproportionately large KL terms relative to math-related tokens, motivating a pointwise cap to prevent these contributions from dominating distillation. Their Section 4.3.3 and Figure 4 show that clipping prevents performance collapse on Qwen3-1.7B. We retain this stabilizer for both OPSD and SRD, holding the distillation objective fixed when comparing their teacher targets. Following the official OPSD implementation, both our OPSD reproduction and SRD optimize a pointwise-clipped surrogate rather than the unmodified teacher–student KL. Capping positive summands can therefore produce negative training losses in either method, as noted in OPSD's official documentation.[6] Section 4.3 evaluates both methods without the cap and shows that SRD retains higher mean AIME24–26 avg@12 than OPSD and the base model at every model size.

Peer sampling and target count. SRD samples four trajectories per problem, uniformly reserves two distinct trajectories as peer contexts, and scores the remaining two. A control trajectory is randomly drawn from those generated for a different problem in the same batch and is shared across the same-problem peer residuals. Thus, 32 problems still yield 128 sampled trajectories and 64 distillation targets per update. The default construction scores four frozen-teacher distributions per target prefix: q_0, q_a, q_b, and q_c. Reusing an in-batch trajectory avoids extra generation, but scoring q_c adds a teacher evaluation. OPSD‡ needs no peer contexts, so it samples and scores two trajectories per problem, yielding the same 64 targets without generating unused trajectories. The default rollout and scored-prefix limits are both 1,024 tokens. The rollout-length ablation extends generation to 2,048 or 4,096 tokens while retaining the 1,024-token scoring limit (Appendix D.3).

C.3 Baseline Details

Table 6 consolidates the controlled configurations for GRPO and TTRL. Both use the same LoRA modules, rank, and scaling as the distillation methods, and share their rollout group, decoding, optimizer, and update budget. Their sole algorithmic difference is the reward label: GRPO checks the normalized final answer against the reference, while TTRL checks it against the current group's normalized majority answer. Our TTRL entry applies this construction to the common training split rather than adapting on benchmark inputs, as marked by † in Table 2. OPSD‡ follows Table 5; Appendix D.5 reports its scaling control separately.

[6] https://github.com/siyan-zhao/OPSD#key-opsd-arguments

Table 6: Controlled configurations for the GRPO and TTRL baselines, which differ only in their reward labels.

Setting	**GRPO**	**TTRL**†
Supervision	Verified reward	Majority pseudo-reward
Sampling	On-policy	On-policy
Problems per update	32	32
Responses per problem	8	8
Trajectories per update	256	256
Maximum response length	8192	8192
Temperature / top-p / top-k	1.1 / 0.95 / 20	1.1 / 0.95 / 20
Optimization	LoRA 64 / 128	LoRA 64 / 128
Learning rate	10^{-6}	10^{-6}
Training budget	300 updates	300 updates
Warmup updates	10	10
Policy-ratio interval	$[0.8, 1.28]$	$[0.8, 1.28]$
Added KL / entropy penalty	None / none	None / none

C.4 Student and Teacher Prompts

The boxes below reproduce the user-message contents used by SRD before applying the model's native chat template. Braced fields are replaced by the current problem and a sampled solution trajectory. Thinking is disabled for the student and all teacher evaluations. The student prompt is used for rollout generation and for scoring both the student distribution and q_0: their input tokens and attention masks are identical, with q_0 evaluated using frozen initial weights. The teacher prompt is used separately for each same-problem distribution q_j and the control q_c, scoring the same target prefix. For q_c, only the {`peer_solution`} field is replaced by a trajectory from another problem; the current problem and prompt wording are retained. The shared prompt explicitly states that the auxiliary solution may or may not address the same problem and does not assume its correctness. For the OPSD baseline, we reuse the original reference-conditioned teacher prompt from Zhao et al. (2026).

Student prompt

```
{problem}

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

Teacher prompt

```
{problem}

An auxiliary solution is provided below. It may or may not address
    the same problem:
=== Auxiliary Solution Begin ===
{peer_solution}
=== Auxiliary Solution End ===

Use the auxiliary solution as potentially useful privileged
    information, but do not assume that its reasoning or final answer
    is correct. Solve the problem using your own independent
    reasoning, checking any information you use from it. Think step by
    step, explore different approaches, and don't be afraid to
    backtrack or reconsider if something doesn't work out:

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

C.5 PASS@12 RESULTS

Table 7 reports pass@12 on AIME24–26 and HMMT25/26 at the corresponding main-table checkpoints. This metric measures the percentage of problems with at least one correct response among twelve samples, complementing the average per-response accuracy reported in the main results. HMMT25 averages the February and November benchmark scores, following Appendix C.1.

Table 7: Pass@12 on AIME and HMMT (%; higher is better). The best score within each model–benchmark block is bold, including ties. Method annotations follow Table 2.

Method	**AIME24**	**AIME25**	**AIME26**	**HMMT25**	**HMMT26**
Qwen3-1.7B					
Base	33.33	23.33	26.67	16.67	27.27
GRPO	40.00	23.33	20.00	18.33	24.24
OPSD‡	33.33	26.67	26.67	**21.67**	24.24
TTRL†	36.67	33.33	20.00	11.67	24.24
SRD (ours)	**46.67**	**40.00**	**33.33**	20.00	**30.30**
Qwen3-4B					
Base	46.67	36.67	46.67	26.67	39.39
GRPO	50.00	36.67	56.67	30.00	30.30
OPSD‡	56.67	43.33	46.67	33.33	36.36
TTRL†	46.67	40.00	50.00	30.00	30.30
SRD (ours)	**76.67**	**70.00**	**73.33**	**45.00**	**42.42**
Qwen3-8B					
Base	53.33	40.00	36.67	28.33	39.39
GRPO	56.67	36.67	46.67	33.33	33.33
OPSD‡	70.00	50.00	56.67	41.67	**42.42**
TTRL†	50.00	40.00	46.67	33.33	33.33
SRD (ours)	**80.00**	**76.67**	**80.00**	**63.33**	**42.42**

SRD achieves the highest pass@12 on all three AIME editions at every model size, showing that its gains extend to problem coverage under a twelve-sample budget.

D Ablation Details

D.1 Ablation Protocol

The core Qwen3-4B ablations in Table 3 are evaluated at the main-table checkpoint and preserve the full method's problem batch, group size, uniform peer sampling, target count, optimization schedule, and evaluation protocol unless that quantity is ablated. Mean-residual, matched-sparsity, signed-mask mean, and full SRD all score q_0, q_a, q_b, and q_c, using the same cross-problem control sampling rule and residuals in Eq. (6). The q_0-reference variant instead sets $d_j(v) = \log(q_j(v)/q_0(v))$, retaining the same q_0-weighted centering and shared-residual aggregation; it requires only q_0, q_a, and q_b. Every residual-based target retains q_0 as its base. The matched-sparsity variant first forms $(\delta_a + \delta_b)/2$. At every scored position, it then uniformly samples, without replacement, exactly as many vocabulary coordinates as are nonzero under the corresponding SRD intersection and zeros the remainder. The random mask is sampled independently of residual sign and magnitude. This matches the per-position support size—and hence the residual zeroing rate—while removing the method's rule for selecting which directions survive. Signed-mask mean instead applies the same all-positive or all-negative test as Eq. (7), but assigns a surviving coordinate the arithmetic mean $M^{-1}\sum_{j=1}^{M}\delta_j(v)$. It therefore preserves the sign mask while removing the conservative minimum-magnitude rule.

D.2 Additional Ablations

Table 8 reports secondary target controls at the main-table checkpoint. The prompt-only teacher receives the problem with instructions to reason independently, explore alternative approaches, and backtrack, without any solution trajectory. Its additional teacher-side instructions distinguish it from q_0, while the absence of a solution context distinguishes it from q_c. This variant tests whether instructions alone can supply useful distillation supervision. Two-peer probability averaging tests whether directly averaging predictions conditioned on the same uniformly sampled peers is sufficient. Section D.3 reports sensitivity to sampling budgets and peer count.

Table 8: Additional Qwen3-4B target-construction ablations. Metrics follow Table 3; Average is the arithmetic mean over AIME24–26.

Variant	AIME24	AIME25	AIME26	Average
Additional target controls				
Prompt-only teacher	7.22	4.17	8.33	6.57
Two-peer probability mean	40.56	29.72	37.50	35.93

The prompt-only teacher uses the following user message:

Teacher prompt

```
Problem: {problem}

Make sure you truly understand the reasoning behind each step. Using
    your own words and independent reasoning, solve the problem above.
    Think step by step, explore different approaches, and don't be
    afraid to backtrack or reconsider if something doesn't work out:

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

D.3 Sampling and Peer-Count Sensitivity

Figure 4 uses Qwen3-4B, the same optimizer schedule and decoding configuration, and mean AIME24–26 avg@12. Panels (a)–(b) track performance throughout training; panel (c) reports performance at the main-table checkpoint.

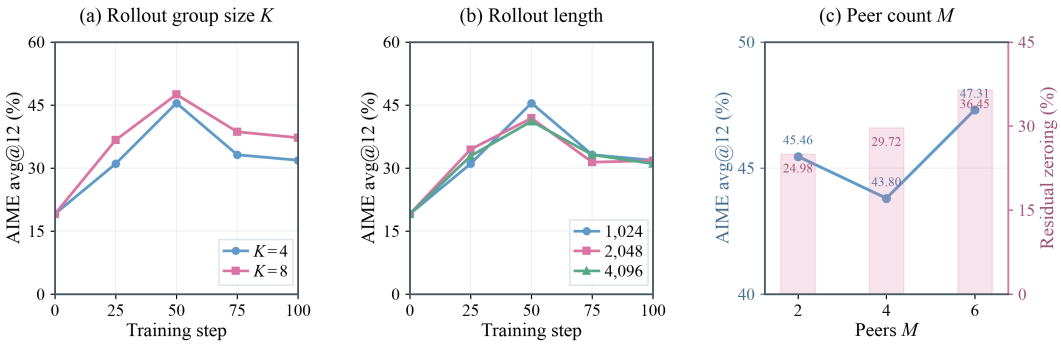

Figure 4: Sampling sensitivity and peer-count analysis on Qwen3-4B. Panels (a)–(b) report mean AIME24–26 avg@12 throughout training while varying rollout group size and rollout length. Panel (c) jointly reports performance at the main-table checkpoint and the training-average residual zeroing rate induced by $M \in \{2, 4, 6\}$ peers, with two scored targets in every setting.

Sampling budgets. Panel (a) varies the total number K of on-policy trajectories while fixing $M = 2$; all remaining $K - 2$ trajectories are scored targets. At the main-table checkpoint, $K = 4$ achieves 45.46 mean avg@12 with half the rollout count of $K = 8$, which achieves 47.50. Panel (b) varies rollout length at $K = 4, M = 2$. The teacher receives full sampled peer trajectories up to the generation limit, but only the first 1,024 target tokens are scored. Cross-problem controls are drawn from the same batch and follow the same generation budget. The 1,024-token setting reaches 45.46, compared with 41.85 and 41.11 for 2,048- and 4,096-token rollouts. These results support the default $K = 4$ and 1,024-token generation budget.

Peer count and residual zeroing. Panel (c) fixes two scored target trajectories and varies $M \in \{2, 4, 6\}$, so each group contains $K = M + 2$ rollouts. The cross-problem control is reused from another group and excluded from M. Equation (7) retains a coordinate only when all peer residuals have the same nonzero sign, taking the smallest magnitude. The residual zeroing rate counts token–vocabulary coordinates zeroed by this sign test, divided by all scored token–vocabulary coordinates. Numerators and denominators are aggregated globally across training before taking their ratio.

As M increases from 2 to 4 and 6, the zeroing rate rises from 24.98% to 29.72% and 36.45%. The default $M = 2$ reaches 45.46 mean avg@12, while $M = 4$ and $M = 6$ reach 43.80 and 47.31, respectively. Two peers already provide effective guidance, while six peers offer a higher score at additional sampling and scoring cost.

D.4 Peer Correctness Composition

Table 9 examines whether peers with incorrect final answers can provide useful supervision. Determining peer correctness requires an extractable final `\boxed{}` answer, which is normalized and checked against the dataset reference. We therefore extend the rollout budget to obtain more completed answers and reduce truncation. In a single offline test on the OPSD training dataset, using a batch of 512 problems with $K = 4$ rollouts per problem, the rollout truncation rate was 43.75% at a 1,024-token limit and 3.03% at a 4,096-token limit. Accordingly, all three conditions use 4,096-token rollouts while retaining the 1,024-token scoring limit.

We set $M = 2$ and evaluate at the main-table checkpoint, with 1,600 groups accepted through dynamic sampling for each condition: two correct peers, one correct and one incorrect peer, or two incorrect peers. The dataset answer is used only to populate these diagnostic batches; the SRD target, number of distillation targets, optimization schedule, and evaluation protocol remain fixed.

At the main-table checkpoint, mean AIME24–26 avg@12 rises from the base model's 19.07 to 40.83 with mixed pairs and 43.43 with two incorrect peers. Thus, peers with incorrect final answers can supply effective supervision, supporting the use of local distributional guidance from unverified solutions. The conditions use different selected problem groups and sampling and training volumes (Table 9); comparisons across compositions are therefore descriptive.

Table 9: Peer-correctness composition analysis on Qwen3-4B at the main-table checkpoint. Every condition uses 4,096-token rollouts, $M = 2$ peers, and 1,600 accepted training groups. Reference answers are used only to determine pair composition. All AIME scores use avg@12, and Average is their arithmetic mean.

Evaluation at the main-table checkpoint

Peer pair	AIME24	AIME25	AIME26	Average
Both correct	40.00	38.61	38.61	39.07
One correct, one incorrect	45.83	37.78	38.89	40.83
Both incorrect	49.44	37.22	43.61	43.43

Observed sampling and training volume

Peer pair	Candidate groups	Accepted rollout tokens	Loss tokens
Both correct	3,088	7,493,105	2,514,815
One correct, one incorrect	8,560	10,593,270	2,944,013
Both incorrect	9,360	10,343,735	2,896,163

Note. Accepted rollout tokens count all four trajectories within selected groups, each capped at 4,096 tokens. Loss tokens count non-padding tokens from the first 1,024 tokens of the two scored trajectories. Lengths of rejected candidates were not logged, so accepted rollout tokens do not measure total sampling cost.

D.5 OPSD Scaling Control

Table 10 evaluates the target-batch adjustment to OPSD‡ at the main-table checkpoint. Original OPSD samples and scores one trajectory per problem ($K = 1$), whereas the matched variant samples two independent trajectories and scores both ($K = 2$). Thus, the comparison changes the number of on-policy targets without introducing peer aggregation or discarding additional rollouts.

Table 10: OPSD target-batch scaling across model sizes (avg@12, %; higher is better). $K = 1$ is the original configuration and $K = 2$ is the target-count-matched variant used in the main comparison.

Model	K	Targets/update	AIME24	AIME25	AIME26
Qwen3-1.7B	1	32	13.33	8.33	12.50
	2	64	14.17	11.39	10.56
Qwen3-4B	1	32	23.61	24.44	24.44
	2	64	22.50	26.39	25.00
Qwen3-8B	1	32	42.22	32.22	33.89
	2	64	40.56	31.94	27.50