PEER-CRITIQUE ON-POLICY DISTILLATION FROM HETEROGENEOUS MODELS FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving an already-capable reasoning model requires repairing its remaining failures. Verifiable rewards can reinforce successful exploration but do not explain how a particular failed trajectory should change; a critique can provide that problem-specific guidance. The model may also fail to diagnose its own error; one remedy is a much larger critic, which adds cost and presumes access to a stronger model. We instead replace the single critic with a panel of same-size peers from different families, whose critiques repair complementary failures: three 3–4B peers repair 41.8–66.0% of failures, exceeding either of two 27–32B critics on every target. PECO (**Pe**er-**C**ritique **O**n-policy Distillation) uses the target's frozen base, conditioned on a failed attempt and a peer critique that enabled a verified repair, as the teacher for on-policy distillation. Already-solved problems enter without critiques, anchoring training to the initial policy to limit forgetting. The construction requires no reference rationale or model larger than the panel; only the verifier accesses the gold final answer. PECO achieves relative gains of 6.4% over the base models and 6.9% over on-policy self-distillation, beats the verifiable-reward baseline on three targets, and outperforms distillation from much larger teachers on all four. PECO-*efficient* prunes the critique pool, retaining 93–102% of PECO's four-benchmark accuracy with 36–44% fewer data-building TFLOPs, and uses $2.1\times$ fewer generation calls than GRPO.

1 INTRODUCTION

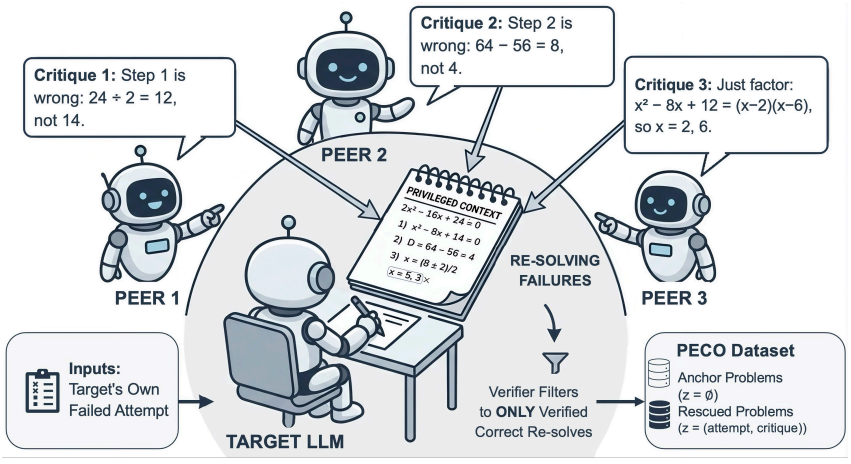

Figure 1: **Peer critique as supervision.** When the target fails, peers of its own scale from other families each critique the failed attempt a; no model sees a reference rationale or the gold final answer, and no larger teacher is used. The target re-solves under each critique c, and a verifier keeps those that turn wrong into right. Rescued problems enter the data with context $z = (a, c)$, already-solved ones with $z = \varnothing$, and the target is trained on this data by on-policy distillation.

Fine-tuning a model on solutions it did not produce teaches the form of an answer more than the ability to find one. This has been reported both for imitating a stronger model (Gudibande et al.,

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.

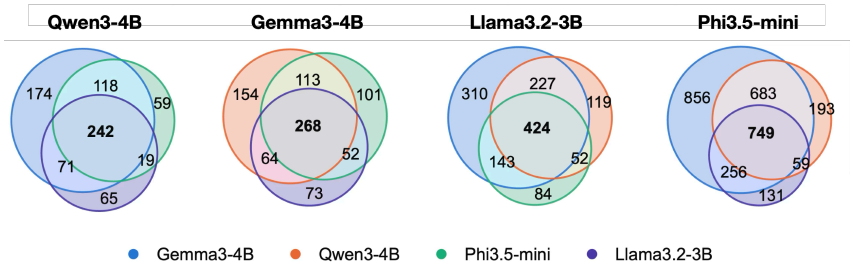

Figure 2: **Complementarity among small peer critics.** Each region counts initially wrong problems repaired by its peer combination. Nonoverlapping regions show failures repaired by only one peer.

2023) and for solution-conditioned training (Kim et al., 2026). In our results, supervised fine-tuning on reference solutions lowers both average and GSM8K accuracy for all four targets, although each solves most GSM8K problems before training (§ 4.2). This loss is a form of catastrophic forgetting, since the update moves the policy away from the base model and prior capability is lost (Kirkpatrick et al., 2017; Shenfeld et al., 2026b). Instead of imitating solutions, the model's failed attempts offer another starting point: a critique can help it repair them.

The cheapest critic is the model itself: it answers, critiques its own reasoning, revises, and trains on the repaired output (Madaan et al., 2023; Shinn et al., 2023; Du et al., 2023; Dhuliawala et al., 2024; Weng et al., 2023; Zelikman et al., 2022; Singh et al., 2023; Kumar et al., 2025; He et al., 2025; Hosseini et al., 2024; Zelikman et al., 2024). The same gaps can hinder diagnosis (Huang et al., 2024; Kamoi et al., 2024; Valmeekam et al., 2023), and self-critique repairs the fewest failures on every target in our comparison (Figure 3). One remedy is a larger critic, which adds cost and presumes access to a stronger model. Comparable peers from different families instead repair failures that no single critic covers. A panel of three such peers repairs 41.8–66.0% of target failures, more than either 27–32B critic on every target; 38–40% are repaired by exactly one peer (Figure 2).

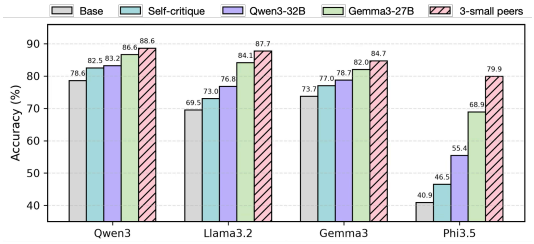

Figure 3: **Critique-and-revise accuracy by critique source.** Only initially wrong answers are revised and correct ones are kept, so bars report overall accuracy. 3-MODEL PEERS exchanges critiques among four models, no model critiquing itself.

The remaining question is how to train on these critiques. Directly fine-tuning on critique text is another form of imitation; in our results it underperforms the base on the two smaller targets (§ 4.2). We instead build on on-policy self-distillation (Snell et al., 2022; Zhao et al., 2026a; Hübotter et al., 2026), in which the student samples its own responses and is trained to match the same model conditioned on a privileged context; on-policy updates of this kind stay close to the base policy and forget less (Shenfeld et al., 2026b; Chen et al., 2025). What fills that context is the open choice: a reference solution, environment feedback, or a single critic (Jin et al., 2026); we ask what a panel of peers adds.

We propose PECO (**Pe**er-**C**ritique **O**n-policy Distillation), which fills that context with a verified peer critique that the verifier has confirmed leads to a correct re-solve (Figure 1). Each model in the panel attempts every training problem and critiques the other models' attempts. The target re-solves failed problems under each peer critique. A verifier keeps critiques that lead to a correct answer. Problems the target already solves are kept with an empty context, which anchors the model to its initial policy.

Contributions. **(i)** Three comparable-scale peers from different families repair more target failures than a larger critic, with complementary repairs (§ 2). **(ii)** We distill verified peer rescues and anchor already-solved problems, requiring only a final-answer verifier (§ 3). **(iii)** Trained on this data, PECO has the best average accuracy on three of four targets and the highest mean over targets, a relative gain of 6.4% over the base models and 6.9% over OPSD (Zhao et al., 2026a) across five math benchmarks (Table 3; § 4). **(iv)** We also propose PECO-*efficient*, which builds its training data with 36–44% fewer TFLOPs and loses at most 2.9 points on any of the four targets (Figure 6; § 4).

Table 1: **Where repairing critiques come from, and when they transfer.** (a) Share of rescued rows by the attempt the critic read. *Base* is the Instruct model's average over five benchmarks from Table 3, which orders the rows. (b) Target repair rate, split by whether the critique repaired the attempt from source peer M_i; a critique that failed on the source attempt rarely transfers.

(a) What the repairing critique was reading.

Target M_k	Base (%)	Critique read (%)		
		Own wrong	Peer wrong	Peer correct
Qwen3-4B	47.9	30	39	31
Gemma-3-4B	40.7	32	32	37
Llama-3.2-3B	32.2	32	25	43
Phi-3.5-mini	30.5	31	10	59

(b) Transfer by whether the critique repaired the attempt from source peer M_i.

Target M_k	M_k repair rate (%)		Ratio	Share (%)
	M_i rep.	M_i not		
Qwen3-4B	47.7	9.6	5.0×	92.8
Gemma-3-4B	29.1	8.6	3.4×	93.7
Llama-3.2-3B	46.9	7.2	6.5×	87.7
Phi-3.5-mini	33.3	3.7	9.0×	82.4

2 Diverse small peers provide useful critiques

Comparable peers from different model families repair complementary subsets of a target's failures: the panel outperforms self-critique and larger critics (Figure 3), and coverage grows with the number of peers (Figure 4).

Setup. Throughout this section we use MATH-train (7,500 problems) (Hendrycks et al., 2021) and four targets from different families: Qwen3-4B (Yang et al., 2025) with thinking disabled, Llama-3.2-3B-Instruct (Grattafiori et al., 2024), Gemma-3-4B-Instruct (Team et al., 2025) and Phi-3.5-mini-Instruct (Abdin et al., 2024). Every critic is one of the other three, so all critics are peers of comparable capacity: each target answers every problem once, each peer critiques the ones it got wrong, and the target re-solves once per critique, counting a problem correct if its first attempt or any re-solve is. On a failed problem the target also re-solves under the critiques the other models received on it, whether the attempt critiqued was right or wrong, excluding only the critiques it wrote itself. Each peer critiques every attempt but its own, so it writes three and the pool holds nine (Table 2). All inference is at temperature 0, so the diversity we measure comes from the critics' perspectives and not from sampling noise.

Different peers rescue different failures. Coverage keeps growing past the second peer because each added peer rescues failures the others miss. Figure 2 decomposes the rescued failures by which peer rescued them: across targets, 38–40% are rescued by exactly one peer. These unique repairs widen the corpus without increasing critic scale.

Which source helps depends on the target. Table 1a sorts the repairing critiques by what the critic was reading: the target's own failed attempt, another model's failed attempt at the same problem, or another model's correct one. All three contribute on every target, so no single source supplies the corpus on its own. The mix is not chosen: whether a critique in the pool read a wrong or a correct attempt is settled per problem, by which peers the verifier scored correct there. It tracks ability all the same. As the target's average accuracy falls, critiques of a peer's wrong attempt drop from 39% to 10% and critiques of a peer's correct attempt rise from 31% to 59%, both monotonically across the four targets, while critiques of the target's own attempt hold near a third throughout. A weaker target needs a critique anchored to a trajectory that worked; a stronger one draws as much from a diagnosis of one that did not.

A critique that does not repair the critiqued attempt rarely transfers. Not every critique in the pool is worth the re-solve pass it needs. Every peer M_i is itself a target in the panel, so a critique of M_i's wrong attempt is re-solved by M_i as well as by M_k, and whether it repaired that source attempt is already known. Table 1b takes these critiques on problems M_k also failed and splits them on that outcome: the first two columns give M_k's repair rate within each group, the third their ratio, and the last the share whose critique did not repair M_i's attempt. That group makes up 82–94% of these critiques and repairs M_k less than a tenth of the time, 3.4–9.0× less often than critiques that repaired M_i's attempt. Since M_i's re-solve is scored anyway, applying the split requires no

additional computation. Keeping only critiques that repaired M_i's attempt reduces the re-solves spent on this source by the same 82–94%. The price is repairs: the dropped group is large enough that it still accounts for 34–81% of the repairs this source yields. PECO-efficient keeps only critiques that repaired M_i's attempt and also drops critiques of a peer's correct attempt; Section 4 reports what this costs in accuracy.

More peers rescue more failures. Figure 4 shows that both accuracy and wrong-to-correct conversion rise monotonically from one to three peers, on every target. The conversion rate in the second panel is the fraction of the target's own failures that becomes usable supervision. Three peers rescue 41.8–66.0% of them, against 26.8–40.8% for a single peer. The gain is largest where the target is weakest: Phi-3.5-mini converts two thirds of its failures and moves from 40.9% to 79.9% accuracy. The models with the least trainable data to begin with are the ones that rescue the most.

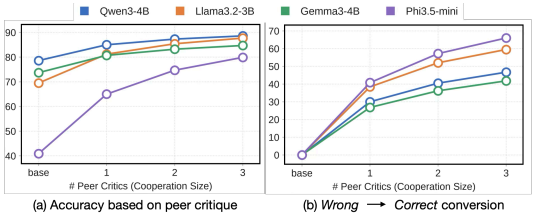

Figure 4: (a) Accuracy on MATH-train and (b) wrong-to-correct conversion rate versus the number of small peer critics. Both rise monotonically from 1 to 3 peers for all four target models.

3 PECO: PEER-CRITIQUE ON-POLICY DISTILLATION

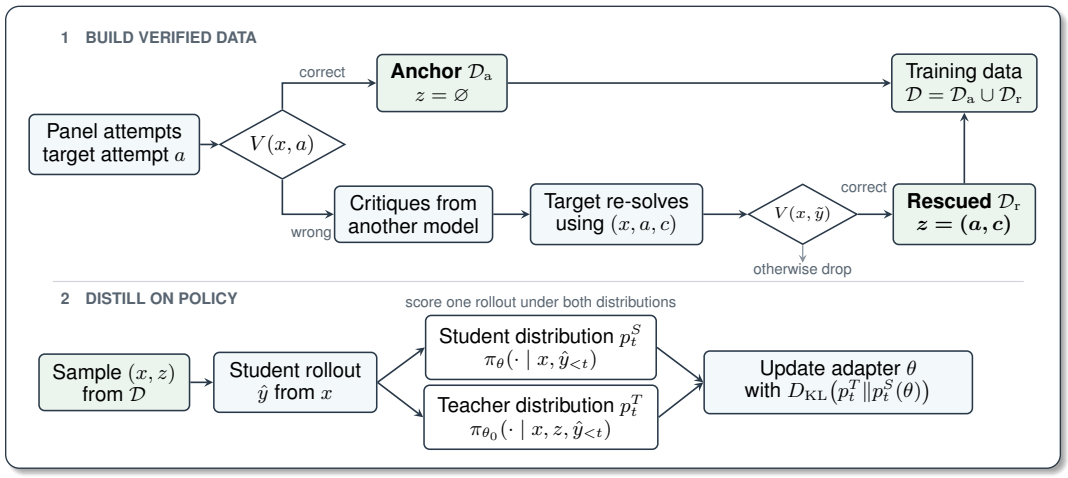

Figure 5: **PECO at a glance:** Green nodes are retained data, blue nodes transformations, and white nodes evaluation. Correct attempts become anchors; failed attempts require a verified peer-guided re-solve, after which the student matches the two distributions on its own rollout (Algorithm 1).

Setup. The panel is K instruction-tuned models $\mathcal{M} = \{M_1, \ldots, M_K\}$ of comparable size, each acting as both solver and critic, and we train one target at a time. Its policy is π_θ, a trainable low-rank adapter θ (Hu et al., 2022) on a frozen base θ_0, and π_{θ_0} is that base. A rule-based verifier $V(x, y) \in \{0, 1\}$ scores an answer y to problem x against a held-out gold final answer the models never see, and only its outcome filters data. Every training example has the same shape, a problem paired with a privileged context, empty or holding a critique according to the verifier, and one objective covers both kinds. We write x for a problem, a for the target's own attempt at it, c for a critique of some attempt at x, and z for that context.

3.1 DATA CONSTRUCTION: RESCUED VS. ANCHOR PROBLEMS

The target produces a single solution a for each training problem x. The verifier $V(x, a)$ decides which of two sets the problem joins. A problem the target solves becomes an *anchor*: it is kept with $z = \varnothing$, so the target is trained to reproduce its own correct reasoning unaided. A problem the target fails is retried under each critique in the pool (Table 2), and if some critique leads to a verified re-solve

the problem becomes *rescued*, paired with that critique as the privileged context z; problems no critique repairs are dropped. We train on the two sets in a 1:1 ratio. Anchors are the scarcer resource for weaker targets, which fail on much of the training set, so when the anchor pool is smaller than the rescued pool we do not subsample to restore the ratio: we keep both sets in full and train on the union.

Rescued set $\mathcal{D}_\mathrm{r}$ (self-failed). Every attempt in the panel is critiqued by the others, so one problem produces a pool of critiques rather than one per peer. Write $a_i \sim M_i(\cdot \mid x)$ for the attempt of M_i, with $a = a_k$ that of the target M_k, and $c_{i \leftarrow j} \sim M_j(\cdot \mid x, a_i)$, $j \neq i$, for the critique of a_i written by M_j; every critique is produced *without seeing the gold final answer*. If $V(x, a) = 0$, the target draws on $\{c_{i \leftarrow j} : j \neq k\}$, every critique on this problem except the ones it wrote itself, over its own failed attempt and over the others alike, correct or not: $(K{-}1)^2$ critiques, nine at $K{=}4$ (Table 2). Excluding self-written critiques is what keeps the context external, since the teacher is the target itself and already holds anything it wrote. A critic is asked for its own corrected solution as well, so c is the part of its response before that correction, truncated at the first heading announcing a corrected solution and at the first boxed answer. The target re-solves under each critique, $\tilde{y} \sim \pi_{\theta_0}(\cdot \mid x, a, c)$, and every pair (x, c) whose re-solve the verifier scores correct enters $\mathcal{D}_\mathrm{r}$ as its own example, with the failed attempt and that critique as the *privileged context* $z = (a, c)$. A problem with several such critiques therefore contributes several rows, and a problem with none is dropped.

Table 2: **The pool one problem produces**, at $K{=}4$ with M_4 as the target. Rows are the attempt a critique reads, columns its author. No model critiques itself ($-$), and the critiques the target wrote are set aside ($\times$), leaving $(K{-}1)^2 = 9$ in all.

	critique written by			
attempt read	M_1	M_2	M_3	M_4
a_1	$-$	peer	peer	$\times$
a_2	peer	$-$	peer	$\times$
a_3	peer	peer	$-$	$\times$
a_4 (target)	own	own	own	$-$

Anchor set $\mathcal{D}_\mathrm{a}$ (self-solved). If $V(x, a) = 1$, the problem needs no help and has no privileged context, $z = \varnothing$. Training runs on $\mathcal{D} = \mathcal{D}_\mathrm{r} \cup \mathcal{D}_\mathrm{a}$, where every example is a pair (x, z) and z is the only thing that distinguishes the two subsets.

PECO-efficient. Each critique in the pool needs a re-solve pass from the target, and most of them do not repay it. PECO-efficient keeps the $K{-}1$ critiques of the target's own failed attempt and, among those written about a peer's failed attempt, only the ones that repaired the peer they were written about; critiques of an attempt that was already correct are dropped in full. The verifier has scored those peer re-solves anyway, since every peer is itself a target of the panel, so the filter is available before a pass is spent rather than after. It discards 82–94% of the critiques written about a peer's wrong attempt (Table 1b), and Section 4.2 (Figure 6) reports the price in accuracy.

3.2 TRAINING

The student is $\pi_\theta(\cdot \mid x)$ and the teacher is $\pi_{\theta_0}(\cdot \mid x, z)$: the *same* network with the adapter disabled and z prepended (Snell et al., 2022). One objective therefore induces two behaviors. On rescued problems the teacher is a critique-aware solver, and distilling it transfers the peer-guided reasoning into π_θ so the student reproduces it *without* a critique at test time. On anchor problems $z = \varnothing$ makes the teacher exactly the base policy on the plain prompt, so the objective there is $D_\mathrm{KL}(\pi_{\theta_0}(\cdot \mid x, \hat{y}_{<t}) \,\|\, \pi_\theta(\cdot \mid x, \hat{y}_{<t}))$ at each token of the student's own rollout, a penalty on moving away from the base where it was already correct. That divergence is the one Shenfeld et al. (2026b) report as the predictor of how much a fine-tuned model forgets.

The student generates its own rollout $\hat{y} \sim \pi_\theta(\cdot \mid x)$, seeing only x, and we match the teacher's next-token distribution along those tokens with a token-clipped forward KL (Zhao et al., 2026a). Writing $p_t^S = \pi_\theta(\cdot \mid x, \hat{y}_{<t})$ and $p_t^T = \pi_{\theta_0}(\cdot \mid x, z, \hat{y}_{<t})$,

$$\theta^\star = \arg\min_\theta \; \mathbb{E}_{(x,z)\sim\mathcal{D}} \, \mathbb{E}_{\hat{y}\sim\pi_\theta(\cdot|x)} \left[\frac{1}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} \sum_{v \in \mathcal{V}} \min\Big(p_t^T(v) \big[\log p_t^T(v) - \log p_t^S(v) \big], \, \tau \Big) \right], \quad (1)$$

where $|\hat{y}|$ is the length of $\hat{y}$; $\mathcal{V}$ is the vocabulary; and τ caps what any one vocabulary entry contributes at any one position. Zhao et al. (2026a) introduce that cap because the per-token divergence is heavy-tailed across the vocabulary. Gradients flow only through p_t^S.

Table 3: Accuracy (%) on math benchmarks across four target models. Critique-FT uses a single Qwen3-32B critic; PECO (ours) uses three small peer critics. *AIME Avg.* pools the two years and *Avg.* is the unweighted mean of GSM8K, GSM-Plus, AIME'24, AIME'25 and Olympiad. **Bold**: best, underline: second best within each model block. Small bold signed values beside *Avg.* give percentage-point changes from the Instruct model.

	Qwen3-4B							**Gemma-3-4B**						
	GSM		AIME			Olymp.	Avg.	GSM		AIME			Olymp.	Avg.
Method	8K	Plus	'24	'25	Avg.			8K	Plus	'24	'25	Avg.		
Instruct	91.8	71.0	16.7	10.0	13.3	49.9	47.9	89.2	68.7	3.3	3.3	3.3	38.9	40.7
SFT (gold)	75.3	55.0	0.0	0.0	0.0	16.0	29.3 **-18.6**	52.2	34.3	0.0	0.0	0.0	9.2	19.1 **-21.6**
KNOWLEDGE DISTILLATION														
KD, Qwen3-32B (w)	91.6	70.0	16.7	13.3	15.0	45.5	47.4 **-0.5**	80.6	58.2	0.0	0.0	0.0	24.9	32.7 **-8.0**
KD, Qwen3-32B (all)	90.4	70.0	13.3	**16.7**	15.0	46.0	47.3 **-0.6**	78.8	58.2	3.3	3.3	3.3	25.5	33.8 **-6.9**
KD, Gemma-3-27B (w)	89.8	69.9	10.0	10.0	10.0	38.0	43.5 **-4.4**	83.2	62.8	6.7	3.3	5.0	31.5	37.5 **-3.2**
KD, Gemma-3-27B (all)	90.1	70.7	6.7	13.3	10.0	42.6	44.7 **-3.2**	82.3	62.5	3.3	0.0	1.7	31.0	35.8 **-4.9**
Critique-FT	**92.6**	70.7	16.7	**16.7**	16.7	45.8	48.5 **+0.6**	87.9	66.3	6.7	13.3	10.0	36.4	42.1 **+1.4**
ON-POLICY														
OPSD	91.7	70.5	13.3	13.3	13.3	47.3	47.2 **-0.7**	89.1	68.5	3.3	3.3	3.3	**40.7**	41.0 **+0.3**
GRPO	91.2	71.0	23.3	13.3	18.3	**50.0**	49.8 **+1.9**	**89.6**	68.3	6.7	6.7	6.7	40.2	42.3 **+1.6**
PECO (ours)	91.9	**71.6**	**30.0**	**16.7**	**23.3**	**50.0**	**52.0** **+4.1**	89.4	**69.2**	**10.0**	**16.7**	**13.3**	39.3	**44.9** **+4.2**

	Llama-3.2-3B							**Phi-3.5-mini**						
	GSM		AIME			Olymp.	Avg.	GSM		AIME			Olymp.	Avg.
Method	8K	Plus	'24	'25	Avg.			8K	Plus	'24	'25	Avg.		
Instruct	80.8	59.3	3.3	0.0	1.7	17.4	32.2	81.1	58.7	0.0	0.0	0.0	**12.6**	30.5
SFT (gold)	61.3	38.0	0.0	0.0	0.0	5.5	21.0 **-11.2**	61.9	44.0	0.0	0.0	0.0	4.7	22.1 **-8.4**
KNOWLEDGE DISTILLATION														
KD, Qwen3-32B (w)	77.8	54.1	3.3	0.0	1.7	17.4	30.5 **-1.7**	80.8	60.5	0.0	**3.3**	1.7	12.3	31.4 **+0.9**
KD, Qwen3-32B (all)	77.7	56.5	3.3	0.0	1.7	16.2	30.7 **-1.5**	80.7	58.1	0.0	**3.3**	1.7	10.5	30.5 **0.0**
KD, Gemma-3-27B (w)	71.3	49.5	0.0	0.0	0.0	13.9	26.9 **-5.3**	75.9	56.2	0.0	0.0	0.0	8.0	28.0 **-2.5**
KD, Gemma-3-27B (all)	72.2	52.2	3.3	0.0	1.7	14.1	28.4 **-3.8**	73.8	53.5	0.0	0.0	0.0	9.5	27.4 **-3.1**
Critique-FT	78.2	57.5	3.3	0.0	1.7	16.5	31.1 **-1.1**	74.4	53.8	**3.3**	0.0	1.7	8.3	28.0 **-2.5**
ON-POLICY														
OPSD	78.2	56.2	**10.0**	0.0	**5.0**	14.7	31.8 **-0.4**	80.6	60.5	0.0	0.0	0.0	11.0	30.4 **-0.1**
GRPO	**82.0**	**60.1**	6.7	0.0	3.3	**17.7**	**33.3** **+1.1**	**82.2**	59.9	**3.3**	0.0	1.7	11.4	31.4 **+0.9**
PECO (ours)	80.6	**60.1**	3.3	**3.3**	3.3	16.9	32.8 **+0.6**	81.2	**61.4**	**3.3**	**3.3**	**3.3**	9.8	**31.8** **+1.3**

4 EXPERIMENTS

4.1 SETUP

Baselines and evaluation. All baselines see the same problems (MATH train dataset (Hendrycks et al., 2021)). **SFT (gold)** (Ling et al., 2017) fine-tunes on the dataset's reference solutions. **KD** (Kim & Rush, 2016) distills rationales from a larger teacher, Qwen3-32B or Gemma-3-27B, either on all problems, marked `all`, or only on the target's failures, marked `w`. The failure-only variants train on the same failure set our rescued set is drawn from. **CFT** (Wang et al., 2025) fine-tunes on a Qwen-3-32B critic's critiques, which amounts to supervised fine-tuning on critique text. **OPSD** (Zhao et al., 2026a) fine-tunes the target using the same model as its teacher, conditioned on a privileged context (the gold solution), in an on-policy manner. **GRPO** (Shao et al., 2024) trains the target by reinforcement learning with the answer verifier as its reward, normalizing each response's reward against a group sampled for the same problem. We report accuracy on GSM8K (Cobbe et al., 2021), GSM Plus (Li et al., 2024), OlympiadBench (He et al., 2024) and AIME 2024/2025 in Table 3.

4.2 MAIN RESULTS

PECO leads on three targets. PECO has the best average accuracy on three targets and the highest mean across them. It gains 4.1 points on Qwen3-4B over its instruct base, and 2.2 over the strongest baseline, GRPO; on Gemma-3-4B it gains 4.2 over the base, and 2.6 over GRPO. PECO also leads on Phi-3.5-mini by 1.3 points over the base. On Llama-3.2-3B, it trails GRPO by 0.5. The on-policy baseline OPSD is beaten on every target. PECO improves accuracy broadly rather than on a single benchmark: averaged over the four targets, it has the highest mean accuracy across GSM8K and GSM-Plus, and it helps most on the hardest problems, raising pooled AIME from 4.6% to 10.8%, 6.2 points over the base and 3.3 over GRPO.

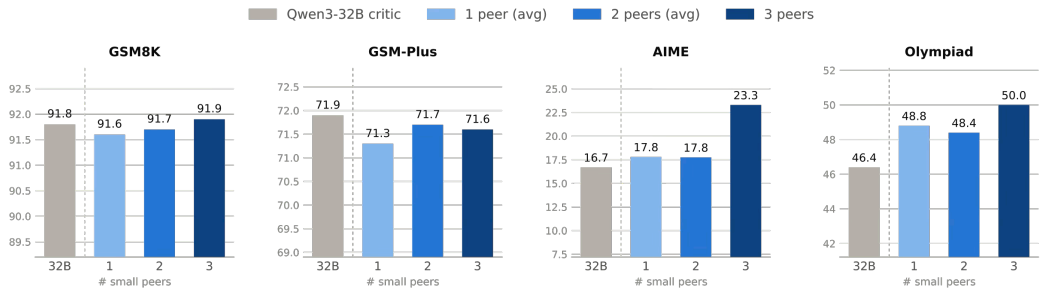

Figure 7: **Scaling the number of peer critics (Qwen3-4B).** Accuracy as the peer panel grows from one to three small critics; the gray bar is PECO with a single Qwen3-32B critic. Bars for one and two peers average over all peer combinations; AIME pools AIME'24 and AIME'25. Accuracy improves with more peers, and three 4B peers surpass the single 32B critic.

Off-policy supervision degrades the student. Fourteen of the sixteen KD entries fall below their base model, and neither the teacher nor the problem selection changes this: rationales from Qwen3-32B and from Gemma-3-27B both fail, and the `-w` variants, trained on the failure set our rescued set is drawn from, are no better than the `-all` variants. Supervision from gold solutions is worse still, with SFT (gold) losing 18.6, 21.6, 11.2 and 8.4 average points; both outcomes match what has been reported for imitating a stronger model (Gudibande et al., 2023) and for solution-conditioned training (Kim et al., 2026). Replacing the solution with a critique is not enough while training stays off-policy: CFT gains 0.6 and 1.4 points on the two 4B targets but loses 1.1 and 2.5 on Llama-3.2-3B and Phi-3.5-mini. On-policy training instead keeps the student near its base: OPSD moves by at most 0.7 points, and GRPO and PECO improve on every target.

Pruning the critique pool costs little accuracy. PECO-efficient keeps a smaller critique pool (§ 3): the $K-1$ critiques of the target's own failed attempt, plus only those critiques of a peer's failed attempt that repaired that peer; critiques of a peer's correct attempt are dropped in full. It builds its data from 56–64% of the compute PECO spends on the same target (Figure 6), and gives up at most 2.9 points of four-benchmark average accuracy (Table 11 in Appendix B). On Qwen3-4B the two are level, at 47.0 against 47.1, and on Llama-3.2-3B the cheaper construction is ahead by 0.4, so on half the panel the discarded critiques were not worth their re-solve passes. The cost falls on Gemma-3-4B, at 2.9 points, which is also the target whose filter separates least, at the 3.4× gap of Table 1b. Phi-3.5-mini has the sharpest filter at 9.0× and still gives up 1.2, so that gap alone does not order the outcomes.

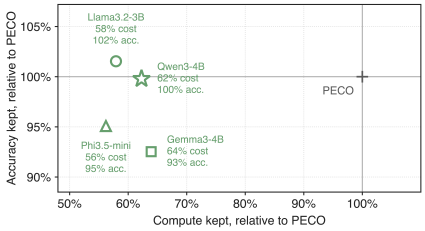

Figure 6: **PECO-efficient against PECO.** Both axes are relative to PECO on the same target: on x the compute spent building the training data, on y the average accuracy over the four benchmarks.

4.3 Ablation studies

More peers help, and beat a larger critic. Figure 7 varies the number of peer critics on Qwen3-4B. Accuracy generally improves as the panel grows, with the largest gains on the hardest benchmarks: AIME improves from 17.8 (one peer) to 23.3 (three peers) and Olympiad from 48.8 to 50.0, while the five-benchmark average goes from 49.5 to 52.0. The 32B critic is competitive on GSM8K and GSM-Plus, but falls behind on AIME (16.7) and Olympiad (46.4), and its average (48.7) is below even a single 4B peer (49.5). Across our controlled pipeline comparison, the heterogeneous small-peer panel outperforms a single much larger critic on all four targets.

Training budget: PECO against GRPO. GRPO samples a group of responses for every problem at every update, whereas PECO generates attempts, critiques and re-solves once, offline, and trains with a single on-policy rollout per row. Averaged over the four targets on GSM8K, OlympiadBench, AIME'24 and AIME'25 (Figure 8), PECO-efficient uses 2.1× fewer generation calls and 1.2× fewer PFLOPs than GRPO while matching its accuracy (+0.4 pp). The full PECO uses about as many calls

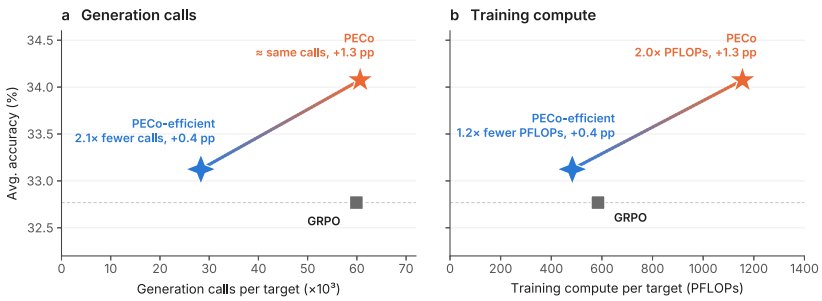

Figure 8: Average accuracy vs. (**a**) generation calls and (**b**) training PFLOPs per target, averaged over four target models. Accuracy is the mean of GSM8K, OlympiadBench, AIME'24 and AIME'25; the line connects the two PECo configurations.

Table 5: Share of evaluation responses (%) containing the re-solving register after one epoch of PECo. A response is counted if it contains any of *previous/original attempt*, *critique*, *reviewer*, *feedback*, *re-derive*, or *from scratch*. Only Phi-3.5-mini is affected, and its base model never produces the revision related tokens. The values in parentheses indicate the share of these responses that re-derive the same answer.

Target	GSM8K	Olymp.	AIME'24	AIME'25
Qwen3-4B	0.0	0.1	0.0	0.0
Llama-3.2-3B	0.0	0.4	0.0	0.0
Gemma-3-4B	0.0	0.0	0.0	0.0
Phi-3.5-mini	0.2	**21.2** (70%)	**33.3** (69%)	**43.3** (69%)
Phi-3.5-mini (base)	0.0	0.0	0.0	0.0

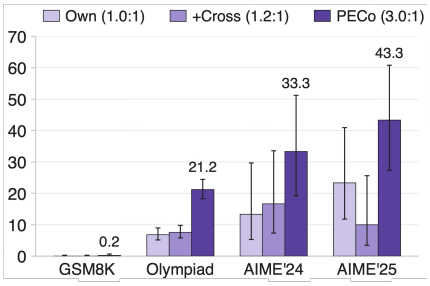

Figure 9: Revision-style token rate on Phi-3.5-mini by critique-source arm, with rescued-to-anchor ratios in the legend and Wilson 95% intervals. Detailed configurations are presented in Appendix B.3.

as GRPO but 2.0× its PFLOPs, since every row requires a teacher pass over the privileged context, and in return gives the largest gain (+1.3 pp). The two variants thus span the budget: PECo-efficient when generation is the bottleneck, and PECo when accuracy is.

Anchor rows prevent forgetting. Training on rescued rows alone still gains on AIME (2.1 points on average), but erodes the benchmarks the base already solves (Table 4): GSM drops by 2.3 points on average and by up to 6.6 on Llama-3.2-3B, and the five-benchmark average falls below Instruct on two of the four targets. Rescued rows come only from problems the base fails, so nothing in that data ties the student to what already works. With anchor rows, whose teacher is the frozen base on the plain prompt, GSM no longer degrades, the AIME gain grows from 2.1 to 6.2 points, and PECo improves over Instruct on every target (Table 3).

Model	GSM	Olymp.	AIME	Avg.
Qwen3-4B	−0.4	−1.8	+1.7	+0.1
Gemma-3-4B	−0.8	+1.3	+5.0	+1.9
Llama-3.2-3B	−6.6	−2.0	0.0	−3.1
Phi-3.5-mini	−1.6	−2.2	+1.7	−0.4
Mean	−2.3	−1.2	+2.1	−0.4

Table 4: Training on rescued rows only (no anchor rows): change from the Instruct model in percentage points. *GSM* averages GSM8K and GSM-Plus; *AIME* pools 2024 and 2025.

Where Phi-3.5-mini falls short: phantom revisions. The teacher's re-solve is written as a second attempt, but the student only ever sees the plain prompt. Three targets almost never reproduce that token style (at most 0.4%; Table 5), whereas Phi-3.5-mini does, more often on harder problems (0.2% on GSM8K, 21.2% on OlympiadBench, 33–43% on AIME). These *phantom revisions* appear after an answer is already written and in about 70% of cases re-derive the same answer: the student has learned what a re-solve looks like, not what it is for. The likely cause is data composition. Phi-3.5-mini solves too few training problems to supply anchors. Its training set therefore has a 1:3 anchor-to-rescued ratio rather than 1:1 (3,068 vs. 9,171 rows). The phantom rate grows with that ratio (Figure 9).

5 Related work

5.1 On-policy distillation and the privileged context

Classical knowledge distillation supervises the student token by token on a fixed corpus of teacher-written sequences, so the student is optimized on trajectories it would not itself produce and compounds error once it decodes on its own (Gu et al., 2024; Agarwal et al., 2024). On-policy distillation samples the training sequences from the student instead and matches the teacher's next-token distribution along them, which keeps the dense per-token signal of distillation on the student's own support (Agarwal et al., 2024). Self-distillation removes the separate teacher: teacher and student are one network under two conditionings, and the teacher reads a privileged context z that the student never sees, so the signal is the difference in what each is shown rather than a gap in parameters (Snell et al., 2022). Recent work composes the two, sampling from the student while the teacher is the same model held at its initial parameters and conditioned on z, and reports it recovering much of the gain of reinforcement learning for a fraction of the rollouts (Zhao et al., 2026a; Shenfeld et al., 2026a).

5.2 Critique-based refinement and training

Natural-language critique was first used at inference time: a model drafts, receives feedback from itself, and revises (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024), with ensembles of critics and multi-agent debate extending the loop to several feedback sources (Mousavi et al., 2023; Du et al., 2023). Inference-time refinement requires the critique-and-revise loop on every query, and its gains vanish when the critic is removed. Intrinsic self-critique is unreliable for reasoning because a model rarely locates an error it was unable to avoid (Huang et al., 2024). A second line therefore moves critique into training, teaching self-correction with multi-turn reinforcement learning (Kumar et al., 2025) or fine-tuning on critiques in place of reference solutions (Wang et al., 2025). Closest to us, recent work treats a critique as the privileged context of an on-policy self-teacher, so that the revision it induces is distilled into a critique-free student (Hübotter et al., 2026); step-aligned critiques are reported to be a stronger context than either binary rewards or reference solutions (Kara & Ersoy, 2026). In these methods the critique comes from one source, typically a much stronger frozen critic; Jin et al. (2026) relax this to a single weak critic, and Byeon et al. (2026) let two models tutor each other. We instead draw critiques from three heterogeneous peers comparable in size to the target. We promote a critique to privileged context only when the verifier confirms that it turns the target's own wrong attempt into a correct one. In our controlled pipeline comparison, this peer panel outperforms a single much larger critic on all four targets.

6 Conclusion

We propose PeCo, peer-critique on-policy distillation, which distills verified critiques from comparable-size peers of other model families into the target while anchoring the problems it already solves to its initial policy. It is the only distillation method in our comparison that improves on the base model on every target, where fine-tuning on gold solutions collapses and distilling a much larger teacher degrades what the model could already do. Across our controlled pipeline comparison, the heterogeneous small-peer panel outperforms a single much larger critic on all four targets. We plan to extend the evaluation beyond mathematical reasoning, where a rule-based verifier is less readily available than it is here.

Limitations. **(i) Cost shifts to data construction.** Training needs no critic, reward model or second set of weights, but every failed attempt must be critiqued and re-solved before it becomes training data. This cost grows with the number of failures, so it is highest for weaker models. **(ii) Evaluation is limited to mathematics.** The construction needs only a verifier for the final answer, so it should extend to domains such as code with unit tests, but we have not tested this. **(iii) Critique quality does not translate uniformly into training gains.** Phi-3.5-mini benefits most from peer critiques at inference, yet gains only $+1.3$ points from training on them; what a model can use in context it does not always internalize.

AI USE STATEMENT

In this work, we used generative AI tools to help implement parts of our method and experiment code, to create the conceptual illustration in Figure 1, to search for related literature, and to polish the writing of the paper. We have not used generative AI tools for research ideation, experimental design, or the interpretation of results.

REPRODUCIBILITY STATEMENT

We make every effort to ensure the reproducibility of our results. The source code for our algorithm is provided as a compressed file in the supplementary materials. Implementation details, hyperparameters, training configurations, and dataset preprocessing steps are described in Appendix A.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

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.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N. M. Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026.

Woohyeon Byeon, Jiwon Jeon, Jeonghye Kim, and Youngchul Sung. Be my tutor: On-policy co-distillation for mutual llm improvement via peer feedback. *arXiv preprint arXiv:2606.14368*, 2026.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 3563–3578, 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.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary LLMs. *arXiv preprint arXiv:2305.15717*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jiayi He, Hehai Lin, Qingyun Wang, Yi R Fung, and Heng Ji. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6405–6421, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

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. In *International Conference on Learning Representations*, 2022.

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.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Can Jin, Jiakang Li, Rui Wu, Eddy Z Zhang, and Dimitris N Metaxas. Weak critics make strong learners: On-policy critique distillation for scalable oversight. *arXiv preprint arXiv:2606.00424*, 2026.

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.

Semih Kara and Oğuzhan Ersoy. The role of feedback alignment in self-distillation. *arXiv preprint arXiv:2606.11173*, 2026.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

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.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

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.

Hyunseok Lee, Seunghyuk Oh, Jaehyung Kim, Jinwoo Shin, and Jihoon Tack. Revise: Learning to refine at test-time via intrinsic self-verification. *arXiv preprint arXiv:2502.14565*, 2025.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984, 2024.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 158–167, 2017.

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.

Sajad Mousavi, Ricardo Luna Gutierrez, Desik Rengarajan, Vineet Gundecha, Ashwin Ramesh Babu, Avisek Naug, Antonio Guillen, and Soumyendu Sarkar. N-critics: Self-refinement of large language models with ensemble of critics. *arXiv preprint arXiv:2310.18679*, 2023.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

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.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026a.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026b.

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Andries Smit, Paul Duckworth, Nathan Grinsztajn, Thomas D Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. *arXiv preprint arXiv:2311.17371*, 2023.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Alex Stein, Furong Huang, and Tom Goldstein. GATES: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Zhiquan Tan and Yinrong Hong. PAINT: Partial-solution adaptive interpolated training for self-distilled reasoners. *arXiv preprint arXiv:2604.26573*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023.

Yubo Wang, Xiang Yue, and Wenhu Chen. Critique fine-tuning: Learning to critique is more effective than learning to imitate. *arXiv preprint arXiv:2501.17703*, 2025.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2550–2575, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, and Yu Hu. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026.

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

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

Yaocheng Zhang, Yuanheng Zhu, Wenyue Chong, Songjun Tu, Qichao Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. π-play: Multi-agent self-play via privileged self-distillation without external data. *arXiv preprint arXiv:2604.14054*, 2026.

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*, 2026a.

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. ROSD: Reflective on-policy self-distillation for language model reasoning across domains. *arXiv preprint arXiv:2605.28014*, 2026b.

APPENDIX

A Implementation details **15**

A.1 Basic experimental setup 15

A.2 Baseline training setup 16

A.3 Prompt templates 17

B Ablation studies **20**

B.1 PECO with a larger critic 20

B.2 On the choice of forward KL 20

B.3 Critique sources ablation 20

B.4 Budget comparison between PECO and GRPO by model 21

C Understanding PECO's gains over OPSD **23**

C.1 Replacing the solution with a critique 23

C.2 Restricting the context to failed problems 23

C.3 Does the content of the critique matter? 23

D Additional related work **24**

D.1 Where the privileged context comes from 24

D.2 The space these sources span 24

D.3 Methods without a conditioned teacher 25

Table 6: Notation used for data construction and training.

Symbol	Meaning
$\mathcal{M} = \{M_1, \ldots, M_K\}$	Panel of K comparable-scale models.
M_k	Target model being trained.
$M_i,\ a_i$	Model whose attempt is critiqued, and its initial attempt on problem x.
$M_j,\ c_{i \leftarrow j}$	Critic peer and its critique of a_i, where $j \neq i$.
$a = a_k,\ \tilde{y}$	Target's initial attempt and its critique-conditioned re-solve.
$V(x, y)$	Rule-based verifier; the only component that accesses the gold final answer.
z	Privileged context: $\varnothing$ for an anchor or (a, c) for a rescued example.
$\mathcal{D} = \mathcal{D}_{\mathrm{a}} \cup \mathcal{D}_{\mathrm{r}}$	Training data formed from the anchor and rescued sets.
$\pi_\theta,\ \pi_{\theta_0}$	Student policy and frozen base policy used as the teacher.
$p_t^S,\ p_t^T$	Student and teacher next-token distributions at position t.
$\mathcal{V},\ \tau$	Vocabulary and pointwise KL clipping threshold.

A Implementation details

A.1 Basic experimental setup

All methods share the same four off-the-shelf instruction-tuned student models, the same source problems, the same LoRA parameterization, and the same evaluation protocol; only the training signal differs. Also, we present the algorithm of PeCo in Algorithm 1.

- **Student models:** Qwen3-4B (Qwen/Qwen3-4B, non-thinking mode), Llama-3.2-3B-Instruct (meta-llama/Llama-3.2-3B-Instruct), Gemma-3-4B-IT (google/gemma-3-4b-it), and Phi-3.5-mini-instruct (microsoft/Phi-3.5-mini-instruct). Throughout, "base" denotes the untouched instruction-tuned checkpoint, not a pre-trained-only model.
- **Source data:** the 7,500-problem MATH training split (`EleutherAI/hendrycks_math`, seven subjects). No benchmark test problem is used for training.
- **Hardware:** 3× NVIDIA RTX 6000 Ada Generation GPUs and 8× NVIDIA RTX A5000 GPUs.
- **Parameter-efficient fine-tuning:** bf16 LoRA (no quantization), rank $r = 16$, $\alpha = 32$, dropout 0.05, applied to every linear projection in the attention and MLP blocks of the language model: `q/k/v/o_proj` and `gate/up/down_proj` for Qwen3-4B, Llama-3.2-3B and Gemma-3-4B, and the fused `qkv_proj`, `o_proj`, `gate_up_proj` and `down_proj` for Phi-3.5-mini. Only the adapter is trained; the frozen backbone with the adapter disabled serves as the teacher.

Table 7: Settings for PeCo. The upper block is training; the lower block is evaluation, and is shared by all methods and all base models.

Setting	Value
Training	
Optimizer	AdamW (PyTorch defaults: $\beta = (0.9, 0.999)$, weight decay 0.01)
Data type	bfloat16
Learning rate	2×10^{-5}, constant (no warmup, no decay)
Epochs	1 (every row seen once)
Gradient clipping	1.0 (global norm)
Rollouts per row	1, on-policy, temperature 1.0, top-p 1.0
Max rollout length	1,024 new tokens
KL direction / clip τ	forward / 1.0 (pointwise)
Evaluation	
Decoding	greedy (temperature 0, top-p 1.0), single sample
Max new tokens	4,096
Grader	`math-verify` 0.9.0

Algorithm 1 Peer-Critique On-policy Distillation (PECo). $c_{m\leftarrow j}$ is written by peer M_j about M_m's attempt, so a critic may read any model's attempt. The attempt itself never reaches the target: z pairs the target's own a with a critique, and only c crosses over.

Require: target model π_{θ_0} (base, $\equiv M_k$) with trainable adapter θ; model pool $\mathcal{M} = \{M_1, \ldots, M_K\}$ (target + peers $\{M_j\}_{j\neq k}$); problems $\{x_i\}$; verifier V; clip τ; lr η; steps N

Phase 1: build data (offline, verifier-filtered)

$\mathcal{D}_{\mathrm{r}} \leftarrow \varnothing$; $\mathcal{D}_{\mathrm{a}} \leftarrow \varnothing$
for each problem x **do**
  $a^{(m)} \sim M_m(\cdot \mid x)$ for all $M_m \in \mathcal{M}$; $a \leftarrow a^{(k)}$ ▷ every model's own attempt
  if $V(x, a) = 1$ **then**
    $\mathcal{D}_{\mathrm{a}} \leftarrow \mathcal{D}_{\mathrm{a}} \cup \{(x,\ z{=}\varnothing)\}$ ▷ easy: anchor problem
  else
    $\mathcal{C}(x) \leftarrow \{c_{m\leftarrow j} \sim M_j(\cdot \mid x, a^{(m)}) \ :\ M_m \in \mathcal{M},\ j \neq k,\ j \neq m\}$
    for each critique $c \in \mathcal{C}(x)$ **do**
      $\tilde{y} \sim \pi_{\theta_0}(\cdot \mid x, a, c)$ ▷ re-solve on the target's *own* attempt a under c
      if $V(x, \tilde{y}) = 1$ **then**
        $\mathcal{D}_{\mathrm{r}} \leftarrow \mathcal{D}_{\mathrm{r}} \cup \{(x,\ z{=}(a, c))\}$ ▷ hard: rescued; keep *all* such critiques
$\mathcal{D} \leftarrow \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{a}}$

Phase 2: on-policy distillation

for $n = 1$ **to** N **do**
  sample $(x, z) \sim \mathcal{D}$
  $\hat{y} \sim \pi_\theta(\cdot \mid x)$ ▷ on-policy rollout: student sees only x
  $p_t^S \leftarrow \pi_\theta(\cdot \mid x, \hat{y}_{<t})$ ▷ student logits along $\hat{y}$ (grad on)
  $p_t^T \leftarrow \pi_{\theta_0}(\cdot \mid x, z, \hat{y}_{<t})$ ▷ teacher = adapter off + context z (stop-grad)
  $z = (a, c)$*: critique-aware teacher ⇒ distill peer reasoning*
  $z = \varnothing$*: teacher = base ⇒ anchor to* θ_0 *(anti-forgetting)*
  $\mathcal{L} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \sum_{v\in\mathcal{V}} \min\Big(p_t^T(v)\big[\log p_t^T(v) - \log p_t^S(v)\big],\ \tau\Big)$ ▷ token-clipped forward KL
  $\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}$
return θ

A.2 Baseline training setup

All baselines train the same LoRA adapter as PECo (rank 16, α=32, dropout 0.05, applied to all linear layers) on top of the frozen Instruct model, and make a single pass over their respective training sets. We denote by $\mathcal{F}$ the set of *stage-1 failures*, i.e., the MATH-train problems (out of 7,500) that the student answers incorrectly under greedy decoding. Table 8 summarizes the settings.

SFT (gold). Standard supervised fine-tuning on the gold solutions of all 7,500 MATH-train problems. We minimize token-level cross-entropy on the target turn only, with the dataset solution (ending in a boxed answer) as the target. Training runs for one epoch.

KD. Sequence-level knowledge distillation from a larger teacher, Qwen3-32B or Gemma-3-27B. The teacher generates one greedy solution per problem (thinking mode disabled), and we use it as the target whether or not it is correct. We consider two variants: **KD-all** trains on all 7,500 problems, while **KD-w** restricts training to $\mathcal{F}$, i.e., only the problems the student originally got wrong. The objective and optimization settings are identical to SFT.

Critique-FT. Following Critique Fine-Tuning (CFT) (Wang et al., 2025), the student learns to critique rather than to imitate solutions. For each problem in $\mathcal{F}$, the input consists of the question together with the student's own stage-1 attempt, and the target is a critique of that attempt produced by Qwen3-32B. As in CFT, critiques are not verified for correctness. The objective and optimization settings are identical to SFT.

Table 8: Training settings of the baselines. All methods share the LoRA configuration of PECo and make one pass over their training set. $\mathcal{F}$: stage-1 failures (MATH-train problems the student answers incorrectly with greedy decoding). CE: token-level cross-entropy; PG: policy gradient.

Method	Loss	Supervision	Problems	Steps
SFT (gold)	CE	gold solution	all 7,500	1 epoch
KD-all	CE	teacher solution	all 7,500	1 epoch
KD-w	CE	teacher solution	$\mathcal{F}$	1 epoch
Critique-FT	CE	Qwen3-32B critique	$\mathcal{F}$	1 epoch
OPSD	forward KL	self, conditioned on gold	all 7,500	1 epoch ($\approx$937)
GRPO	PG	binary verifier reward	7,488	1 epoch (234)

OPSD. On-policy self-distillation (Zhao et al., 2026a), in which the teacher is the frozen Instruct model itself, given privileged access to the gold reference solution in its context. For each training row, the student samples one rollout (temperature 1.0, top-p 1.0, $\leq$1,024 new tokens), and we minimize the per-token forward KL divergence $\mathrm{KL}(\pi_T \,\|\, \pi_S)$ over the student's own rollout tokens, clipping each token's term at 1.0. We train on all 7,500 problems for one epoch ($\approx$937 steps) with gradient clipping at 1.0.

GRPO. Reinforcement learning with group-relative policy optimization (Shao et al., 2024). For each prompt we sample 8 rollouts (temperature 1.0, top-p 1.0) and assign a binary reward $R \in \{0, 1\}$ by checking the boxed answer with MATH-VERIFY, with no format bonus. The advantage is the sequence-level, group-centered reward $A = R - \bar{R}_{\text{group}}$, without standard-deviation scaling; groups in which all rewards are equal carry no signal and are dropped. Because we perform a single on-policy update per batch, no PPO clipping is needed, and we use neither a KL penalty nor a reference model. Training runs for 234 steps (one pass over 7,488 prompts) with gradient clipping at 1.0.

A.3 Prompt templates

A.3.1 Prompt for data construction

All three stages share the same placeholder convention: `{question}` is the problem statement, `{peer_key}` is the identifier of the peer model whose attempt is being reviewed, `{peer_attempt}` / `{previous_attempt}` is the full text of a prior solution, and `{critique}` is the reviewer output from the critique stage. Every stage terminates with the same `\boxed{}` answer format so that answers can be extracted automatically and compared across stages.

Stage 1: Initial solution. Each model independently produces a first attempt at the problem.

Stage 1 — Initial solution prompt

```
[system]
You are a careful math solver. Reason step by step, then end with the requested format.

[user]
Solve the following problem.

### PROBLEM
{question}

Reason step by step. At the very end, write the final answer in exactly this LaTeX form:
\boxed{answer}
```

Stage 2: Peer critique. A model is shown exactly one peer attempt, asked to identify its mistakes, and to produce its own corrected solution.

Stage 2 — Critique prompt

```
[system]
You are a careful math critic. You will see a problem and exactly one peer model attempt.
Identify mistakes in that attempt and produce your own corrected answer.

[user]
### PROBLEM
{question}

### PEER_ATTEMPT
--- attempt by {peer_key} ---
{peer_attempt}

Critique the peer attempt above, then produce your own corrected solution. Reason step by
step.
At the very end, write the final answer in exactly this LaTeX form:
\boxed{answer}
```

Stage 3: Revision. The original solver re-derives the solution from scratch, conditioned on both its previous attempt and the reviewer's critique. The instruction not to trust the previous attempt blindly is included to discourage the model from simply copying its earlier answer.

Stage 3 — Revision prompt

```
[system]
You are a careful math solver. Follow the requested output format.

[user]
{question}

Below is a previous attempt at solving this problem along with a reviewer's critique.
Take the critique into account, do NOT trust the previous attempt blindly,
and re-derive the solution from scratch.

### PREVIOUS_ATTEMPT
{previous_attempt}

### REVIEWER_CRITIQUE
{critique}

Now provide step-by-step reasoning. At the very end, write the final answer
in exactly this LaTeX form:
\boxed{answer}
```

A.3.2 Prompt for training & evaluation

During on-policy distillation the student and the teacher are the same network and differ only in the prompt they read (Algorithm 1). The student always receives the direct-solve template (Prompt T1), on anchor and rescued rows alike; its rollout $\hat{y}$ is sampled from this prompt, and the same template is used at evaluation. T1 is deliberately minimal: unlike the Stage 1 prompt, it carries no `### PROBLEM` header and no extra role framing, so the student is trained and evaluated under the same context. The teacher's prompt depends on the row. On an anchor row ($z = \varnothing$) the teacher reads the same T1 prompt, so it is exactly the frozen base policy on the plain prompt. On a rescued row ($z = (a, c)$) it reads the critique-conditioned revision template (Prompt T2), which is identical to the Stage 3 revision prompt, so the teacher's next-token distributions along $\hat{y}$ are computed under the same context in which the verified re-solve was produced. T2 is used only for these teacher distributions during training and never at evaluation.

T1 — Direct solution (student; teacher on anchor rows; evaluation)

```
[system]
You are a careful math solver. Follow the requested output format.

[user]
{question}

Solve the problem with step-by-step reasoning. At the very end, write the final answer
in exactly this LaTeX form:
```

```
\boxed{answer}
```

T2 — Critique-conditioned revision (teacher on rescued rows; training only)

```
[system]
You are a careful math solver. Follow the requested output format.

[user]
{question}

Below is a previous attempt at solving this problem along with a reviewer's critique.
Take the critique into account, do NOT trust the previous attempt blindly,
and re-derive the solution from scratch.

### PREVIOUS_ATTEMPT
{previous_attempt}

### REVIEWER_CRITIQUE
{critique}

Now provide step-by-step reasoning. At the very end, write the final answer
in exactly this LaTeX form:
\boxed{answer}
```

B ABLATION STUDIES

B.1 PECO WITH A LARGER CRITIC

To compare the peer panel with a stronger critic, we replace the three small peer critics with a single Qwen3-32B critic while keeping the rest of the training pipeline unchanged. Table 9 compares the two variants. PECO with the 32B critic matches or improves upon the Instruct model across all four targets, confirming that critique-guided on-policy training remains effective with a single strong critic. The small-peer variant has higher average accuracy on every target, with the largest gaps on Qwen3-4B and Gemma-3-4B because of its stronger AIME results. The controlled comparison therefore favors the heterogeneous small-peer panel over a single 32B critic on every target.

Table 9: Accuracy (%) of PECO with a single Qwen3-32B critic versus PECO (ours) with three small peer critics. **Bold**: better of the two within each model.

		GSM		AIME				
Model	Method	8K	Plus	'24	'25	Avg.	Olymp.	Avg.
Qwen3-4B	PECO (32B critic)	91.8	**71.9**	13.3	**20.0**	16.7	46.4	48.7
	PECO (ours)	**91.9**	71.6	**30.0**	16.7	**23.3**	**50.0**	**52.0**
Gemma-3-4B	PECO (32B critic)	88.5	67.8	3.3	3.3	3.3	**39.9**	40.6
	PECO (ours)	**89.4**	**69.2**	**10.0**	**16.7**	**13.3**	39.3	**44.9**
Llama-3.2-3B	PECO (32B critic)	**81.3**	59.9	3.3	0.0	1.7	16.8	32.3
	PECO (ours)	80.6	**60.1**	3.3	**3.3**	**3.3**	**16.9**	**32.8**
Phi-3.5-mini	PECO (32B critic)	**82.0**	58.9	3.3	0.0	1.7	**11.4**	31.1
	PECO (ours)	81.2	**61.4**	3.3	**3.3**	**3.3**	9.8	**31.8**

B.2 ON THE CHOICE OF FORWARD KL

$D_{\mathrm{KL}}(p^T \| p^S)$ is mass-covering. On anchor problems it forces the student to keep covering the base policy's modes, which protects easy-task accuracy. On rescued problems it spreads probability toward the corrected reasoning rather than collapsing to a single mode. We compare forward and reverse KL on Qwen3-4B in Table 10.

Table 10: Forward KL beats reverse KL (Qwen3-4B-Instruct). It wins on all four benchmarks, improving the average gain over the base model from $+2.33$ to $+5.05$.

	GSM8K	Olympiad	AIME24	AIME25	Avg. Δ vs. Base
Reverse-KL	91.5	49.6	23.3	13.3	$+2.33$
Forward-KL	**91.9**	**50.0**	**30.0**	**16.7**	$\mathbf{+5.05}$

B.3 CRITIQUE SOURCES ABLATION

Section 3 admits any critique the panel produces, whether the critic was looking at the target's own attempt or at another model's. Table 12 asks whether that breadth does any work. Writing $c_{m \leftarrow j}$ for the critique M_j writes about M_m's attempt at x, the three arms admit nested critique sets: OWN takes $c_{k \leftarrow j}$, +CROSS adds $c_{m \leftarrow j}$ for an M_m that also failed x, and PECO further adds $c_{m \leftarrow j}$ for an M_m that solved it, and are otherwise identical. The privileged context is $z = (a, c)$ throughout, with a the target's own attempt: only the critique's provenance changes. Row counts are an outcome rather than a knob, since widening the source set is what admits more verified rescues; the full set roughly triples the rescued rows on every target.

PECO has the best average on three of four targets and the best mean over targets, 34.1 against 32.5 for OWN and 32.1 for +CROSS; Llama-3.2-3B is the exception, where OWN leads by 0.7. The cleaner comparison is between the two wider arms: PECO beats +CROSS on all four targets, by 0.6 to 5.1

Table 11: Accuracy (%) of PECO and PECO-efficient on four math benchmarks. Both use three small peer critics.

	Qwen3-4B						*Gemma-3-4B*					
	Math		AIME				Math		AIME			
Method	GSM8K	Olymp.	'24	'25	Avg.	Avg.	GSM8K	Olymp.	'24	'25	Avg.	Avg.
PECO (ours)	91.9	50.0	30.0	16.7	23.3	47.1	89.4	39.3	10.0	16.7	13.3	38.8
PECO- efficient (ours)	91.9	49.4	26.7	20.0	23.3	47.0	88.9	41.2	6.7	6.7	6.7	35.9
	Llama-3.2-3B						*Phi-3.5-mini*					
	Math		AIME				Math		AIME			
Method	GSM8K	Olymp.	'24	'25	Avg.	Avg.	GSM8K	Olymp.	'24	'25	Avg.	Avg.
PECO (ours)	80.6	16.9	3.3	3.3	3.3	26.0	81.2	9.8	3.3	3.3	3.3	24.4
PECO- efficient (ours)	81.1	17.7	3.3	3.3	3.3	26.4	81.6	11.3	0.0	0.0	0.0	23.2

points, whereas +CROSS against OWN splits two and two and moves in both directions (-3.8 on Qwen3-4B, $+3.8$ on Gemma-3-4B).

Table 12: Critique sources ablation. All three arms are trained with the same objective and schedule (1 epoch, anchor:rescued = 1:1), and differ only in which critiques may enter the privileged context z. Their row counts therefore differ, by a factor of 3.3 on Qwen3-4B, and are given in the last three columns. OWN keeps only critiques written about the target's own failed attempt; +CROSS adds critiques written about *another* model's failed attempt on the same problem; PECO additionally admits critiques of another model's *correct* attempt. In every arm the attempt shown inside z is the target's own; only the provenance of the critique changes. Avg. is the unweighted mean over the four benchmarks. Bold marks the best entry within each model block.

							# Training rows		
Model	Arm	GSM8K	Olymp.	AIME'24	AIME'25	Avg.	Rescued	Anchor	Total
Qwen3-4B	Exp1 (own)	91.5	48.5	23.3	**20.0**	45.8	726	726	1,452
	Exp2 (+cross)	91.6	49.9	13.3	13.3	42.0	1,649	1,649	3,298
	PECO	**91.9**	**50.0**	**30.0**	16.7	**47.1**	2,384	2,384	4,768
Gemma-3-4B	Exp1 (own)	88.3	39.5	3.3	6.7	34.4	834	834	1,668
	Exp2 (+cross)	88.7	**40.8**	**13.3**	10.0	38.2	1,668	1,668	3,336
	PECO	**89.4**	39.3	10.0	**16.7**	**38.8**	2,628	2,628	5,256
Llama-3.2-3B	Exp1 (own)	**80.7**	16.2	**6.7**	**3.3**	**26.7**	1,400	1,400	2,800
	Exp2 (+cross)	80.1	16.3	3.3	0.0	24.9	2,465	2,465	4,930
	PECO	80.6	**16.9**	3.3	**3.3**	26.0	4,336	4,336	8,672
Phi-3.5-mini	Exp1 (own)	81.7	10.4	0.0	0.0	23.0	2,820	2,820	5,640
	Exp2 (+cross)	**82.2**	**10.7**	0.0	0.0	23.2	3,767	3,068†	6,835
	PECO	81.2	9.8	**3.3**	**3.3**	**24.4**	9,171	3,068†	12,239

† Anchors are capped by the number of problems the target already solves (3,068 for Phi-3.5-mini), so the 1:1 ratio cannot be met once the rescued set grows beyond it.

B.4 BUDGET COMPARISON BETWEEN PECO AND GRPO BY MODEL

Figure 10 reports the per-target budget behind Figure 8. PECO-efficient uses fewer generation calls than GRPO on every target, with the largest saving on Qwen3-4B ($3.0\times$ fewer calls, $+2.6$ pp); Phi-3.5-mini is the only target where it costs more PFLOPs than GRPO.

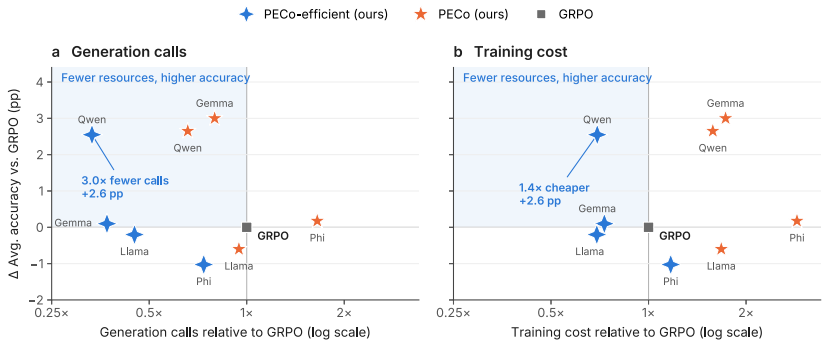

Figure 10: Accuracy against generation calls (**a**) and training PFLOPs (**b**), relative to GRPO shown as a square. Accuracy is the mean of GSM8K, OlympiadBench, AIME'24 and AIME'25.

C UNDERSTANDING PECO'S GAINS OVER OPSD

PECO and OPSD (Zhao et al., 2026a) share the same training objective: the student samples its own rollout, and the teacher is the frozen initial policy conditioned on a privileged context z that the student does not see. Because the teacher starts with the same weights as the student, it can provide a useful signal only through z. The two methods differ in what z contains and where it is applied. OPSD sets z to a reference solution on every problem, whereas PECO sets it to a verified peer critique on problems the target fails and leaves it empty on problems the target already solves. Table 3 reports PECO ahead of OPSD on all four targets. We first discuss one explanation for each of the two differences, and then examine an alternative: that critiques help only because they carry parts of a solution, in which case changing the content of z would not matter.

C.1 REPLACING THE SOLUTION WITH A CRITIQUE

Prior work attributes the degradation in self-distillation to the solution in the teacher's context rather than to the on-policy objective. Kaur et al. (2026) fix the student and the token budget and vary only the teacher's context. On Qwen3-1.7B, on-policy distillation from a larger teacher that sees only the prompt raises avg@16 from 0.372 to 0.392; adding a gold demonstration to that teacher's context lowers it to 0.350 (paired difference -0.042, interval $[-0.065, -0.019]$), and self-distillation with a full gold demonstration lowers it further to 0.308. PECO removes this component, though not completely: in 31–59% of the rescued rows the critique was written about a correct attempt (Table 1a), and we do not measure how much of that solution the critique retains. Since Kaur et al. (2026) use different students, benchmarks and budgets, and do not condition any teacher on a critique, their result motivates the substitution but does not measure its effect in our setting. The substitution also makes a heterogeneous panel possible. The critique enters the teacher's prompt as text, and both p_t^S and p_t^T are computed by the target's own network, so the two distributions share one vocabulary. Distilling directly from the peers' distributions would instead require reconciling four different tokenizers.

C.2 RESTRICTING THE CONTEXT TO FAILED PROBLEMS

Armandpour et al. (2026) measure, per token and without training, whether a teacher's distillation gradient points toward the student's success, and find it reliably aligned on the student's failing rollouts but noisy on its correct ones. PECO's verifier splits the data along the same line, although at the level of problems rather than rollouts. Rescued rows come from problems where the target's initial attempt was wrong, where the teacher's signal is expected to be reliable. On anchor rows $z = \varnothing$, so the teacher is the base policy on the plain prompt and the objective penalizes the student's divergence from the base. Shenfeld et al. (2026b) identify this divergence as a predictor of forgetting, and it is zero at initialization by construction. Anchor rows make up half of the training set on three targets, where they are subsampled to match the rescued rows, and a quarter on Phi-3.5-mini, which solves too few problems to match them (Table 12). OPSD has no counterpart to these rows: its teacher reads a solution on every problem, including those the target already solves. The alignment result is measured per rollout on a single student–teacher pair, so it is consistent with this routing but does not establish its benefit. The direct evidence in our setting is the rescued-only ablation (Table 4): without anchor rows, GSM accuracy drops on every target and the five-benchmark average falls below the base model on two of them.

C.3 DOES THE CONTENT OF THE CRITIQUE MATTER?

Neither explanation above separates the content of the critique from the objective that delivers it. If PECO helped mainly because critiques leak solutions into z, removing the critiques closest to a solution should remove most of the gain. PECO-efficient performs approximately this removal: it discards every critique of an attempt that was already correct, the only part of the pool written about a working solution. It loses at most 2.9 points relative to PECO and keeps its four-benchmark gain over the base model on three of the four targets (Table 11); the gain disappears only on Phi-3.5-mini, where it was 1.0 point. This bounds the effect rather than isolating it, because PECO-efficient also discards most critiques of peers' failed attempts. A second, weaker piece of evidence comes from changing the critic while keeping the objective and the rest of the pipeline fixed (Table 9): replacing the three peers with a single Qwen3-32B critic lowers accuracy on all four targets, by 3.3–4.3 points

on the two 4B targets but by less than one point on Llama-3.2-3B and Phi-3.5-mini. Finally, Stein et al. (2026) condition the teacher on a reference document instead of a solution and report gains, whereas Kaur et al. (2026) find that a gold demonstration degrades the student; this is the same kind of substitution applied to a different context.

D Additional related work

D.1 Where the privileged context comes from

On-policy self-distillation methods differ mainly in what supplies the privileged context z. We group them by the source of z.

From the dataset. Zhao et al. (2026a) set z to a reference solution and keep the teacher fixed at the initial policy, on the premise that a model given a correct solution can explain it and thereby teach its unconditioned self. Tan & Hong (2026) reveal only part of the solution and prove that the withheld part should grow as the student's alignment with the solution improves. Shenfeld et al. (2026a) condition the teacher on a demonstration and report that this retains prior capabilities that fine-tuning on the same demonstrations loses.

From the model's own rollouts. Hübotter et al. (2026) condition on environment feedback or, when none is available, on a successful rollout from the same batch, and update the teacher within a trust region instead of freezing it. Yu et al. (2026) condition on other rollouts of the same model and find that successful rollouts provide most of the signal. Zhao et al. (2026b) pair a failed rollout with a correct one from the same group, condition the teacher on their comparison rather than on the solution alone, and restrict the loss to the tokens after the located error. Without environment feedback, all three methods require the model to solve the problem at least once.

From outside the model. Penaloza et al. (2026) obtain z from a stronger model, grade it by how much of the answer it reveals, and train the conditioned teacher with its own objective so that its outputs remain reachable by the student. Stein et al. (2026) use a reference document supplied with the question. Because a single reading of the document can be wrong, they sample several independent readings and distill only where enough of them agree.

Terminology. Several of these works use plural terms for a single model. Yu et al. (2026) call sibling rollouts of one model "peers", Zhao et al. (2026b) have one model reflect on its own rollouts, and Zhang et al. (2026) describe one base model playing three roles as "multi-agent". None of them uses more than one model. In this paper, a *peer* is a separate model of comparable size from a different model family.

D.2 The space these sources span

Table 13 places these methods on two axes: the source of z (rows) and how much z reveals about the answer (columns). The columns describe only what the context contains, so a correct rollout and a reference solution both fall under *the answer* and differ only in their row. The column *that it failed* corresponds to environment feedback, which reports a failure without analyzing the reasoning and, in mathematical reasoning, amounts to a single bit. We omit learning from outcome rewards alone, since it involves no conditioned teacher. The last column is a property of the source rather than a third axis: it records whether constructing z requires an answer to exist before the pipeline runs, as a reference solution in the dataset, a document containing one, a stronger model able to produce one, or a correct rollout of the model itself. Judging whether an attempt is correct is different from knowing the answer: a verifier only returns correct or incorrect and never places an answer in any model's context.

PECO occupies two cells of the bottom row, and the verifier, not a design choice, decides which cell each problem falls into. A problem the target already solves receives no context; a problem it fails receives a critique. This placement follows from the construction rather than from a measurement: we remove the critic's corrected solution and final answer from the critique, but we do not measure how much of the solution the remaining text reveals. This matters most for critiques of attempts that

Table 13: Where a privileged context comes from, against how much it says about the answer. † withholds an adjustable part of the answer, so it moves between the last two columns.

its source	*what the context says about the answer*				*an answer must exist*
	nothing	that it failed	where it failed	the answer	
the dataset	–	–	–	Zhao et al. (2026a), Shenfeld et al. (2026a), Tan & Hong (2026)†	yes
the environment	–	Hübotter et al. (2026)	–	–	no
a reference document	–	–	–	Stein et al. (2026)	yes
a stronger model	–	–	Penaloza et al. (2026)	Penaloza et al. (2026)	yes
the model itself, same rollout group	–	–	Zhao et al. (2026b)	Yu et al. (2026), Hübotter et al. (2026)	yes
the model itself, in another role	–	–	–	Zhang et al. (2026)	yes
heterogeneous model family	**PECO**, anchor set	–	**PECO**, rescued set	–	**no**

were already correct, which supply 31–59% of the rescued rows (Table 1a) and plausibly reveal more of a solution than critiques of failed attempts. PECO-efficient drops these critiques to save compute (Section 2), so the cheapest variant is also the one whose context reveals the least about the solution.

The last column is **no** for PECO because building z requires no answer: the verifier only judges whether an attempt is correct, and no model ever sees the gold final answer. When a critique is written about a peer's correct attempt, that attempt is produced by the panel during data construction rather than supplied in advance. The full construction does not select critiques by whether the critiqued attempt was correct; PECO-efficient uses this information only to prune the pool. The table makes no claim about how much each kind of answer helps. For example, Kaur et al. (2026) report that in thinking models a full demonstration shortens long-budget rollouts more than a final answer alone.

D.3 METHODS WITHOUT A CONDITIONED TEACHER

Three lines of work fall outside this taxonomy because no teacher in them reads information that the student lacks. Test-time self-correction has a model critique and revise its own output without any parameter update (Madaan et al., 2023; Gou et al., 2024; Lee et al., 2025; Weng et al., 2023; Dhuliawala et al., 2024; Shinn et al., 2023; Du et al., 2023; Liang et al., 2024; Smit et al., 2023); it relies on the model diagnosing its own mistakes, which is unreliable when the error stems from a gap in the model's own knowledge (Huang et al., 2024; Kamoi et al., 2024; Valmeekam et al., 2023). STaR- and ReST-style methods train on self-generated rationales filtered by outcome (Zelikman et al., 2022; 2024; Singh et al., 2023), as does reinforcement learning for self-correction (Kumar et al., 2025); the training signal therefore comes from a verifier rather than a teacher. Critique fine-tuning uses teacher-written critiques as training targets rather than as the teacher's context (Wang et al., 2025).