ON-POLICY DISTILLATION OF LANGUAGE MODELS BY REPAIRING THE FUTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) supervises student-generated trajectories, but local token corrections do not guarantee better final outcomes because they can redirect the entire autoregressive continuation. We propose **REPAIR-OPD**, which identifies candidate repair points from teacher confidence and teacher–student disagreement, applies only a local teacher correction, and lets the student regenerate the remaining trajectory. Only Judge-verified **wrong-to-correct** repairs are used for learning. Across five mathematical-reasoning and two out-of-domain benchmarks, REPAIR-OPD achieves an overall mean of **54.55%**, outperforming the strongest OPD baseline by **4.72%**, and remains effective across smaller Qwen and cross-family Llama settings. In a matched one-round efficiency test on 1,000 source questions, REPAIR-OPD requires $1.36\times$ OPD's end-to-end training time, while its seven-benchmark mean response length reaches $1.69\times$ that of OPD, indicating a **controlled cost–performance trade-off**. Our results suggest that effective OPD should optimize not only local corrections, but the **future trajectories they induce**. Code is available at the anonymous repository.

1 INTRODUCTION

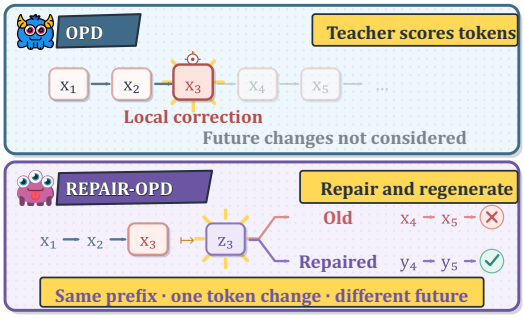


Figure 1: OPD vs. REPAIR-OPD. OPD supervises the observed trajectory, whereas REPAIR-OPD corrects a token and repairs its future.

On-policy distillation (OPD) trains a student language model with teacher supervision on trajectories sampled from the student's current policy (Gu et al., 2024; Agarwal et al., 2024). By supervising student-visited prefixes, OPD allows the student to learn from mistakes exposed by its own rollouts rather than only from fixed teacher responses. However, token-level supervision on an observed rollout does not determine whether a local correction improves the completed response, because changing one autoregressive decision can redirect the entire continuation (Lee et al., 2026). Effective OPD should therefore look beyond local token feedback and repair the student's **autoregressive future** when it leads to an incorrect outcome.

The original OPD paradigm distills teacher supervision directly on student-generated trajectories (Gu et al., 2024; Agarwal et al., 2024). Subsequent work follows two directions. First, **supervision design** determines how teacher feedback is constructed and used: some methods introduce richer contexts or rollout structures (Zhao et al., 2026; Ye et al., 2026; Yu et al., 2026c; Tan & Hong, 2026; Zhang et al., 2026b), while others calibrate, reweight, or reformulate the training objective (Jin et al., 2026; Jang et al., 2026; Zheng et al., 2026; Yang et al., 2026; Fu et al., 2026; Xu et al., 2026a; Ko et al., 2026; Oh et al., 2026; Ke et al., 2026). Second, **token selection** determines where teacher feedback is applied, prioritizing early prefixes, student uncertainty and teacher confidence, uncertain or overconfident predictions, teachable disagreements, agreement-based trust regions, and decision-supporting tokens (Zhang et al., 2026a; Kim & Baek, 2026; Xu et al., 2026b; Wang et al., 2026; Xing et al., 2026; Xiao et al., 2026). These methods refine OPD while retaining its core setting: teacher supervision on trajectories sampled from the student's current policy.

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.

However, autoregressive generation is **path-dependent**: replacing one token changes the context (Xu et al., 2025), so supervision on the original suffix no longer applies to the regenerated future. Standard OPD methods typically optimize feedback on observed student trajectories, but do not make the student-generated future an outcome-verified learning target, as shown in Figure 1.

To address this limitation, **we introduce REPAIR-OPD**, which combines local teacher intervention, student-generated suffix regeneration, and outcome verification: teacher–student disagreement identifies candidate repair points, a bounded breadth-first search explores alternatives within a finite repair budget, and a fixed Judge retains only wrong-to-correct trajectories for learning.

To assess its effectiveness, **we evaluate REPAIR-OPD** on five mathematical-reasoning and two out-of-domain benchmarks. With `Qwen3-8B` (Yang et al., 2025) as teacher and `Qwen3-4B` (Yang et al., 2025) as student, the three-seed mean is **54.55%**, **4.72%**↑ over the strongest baseline and **8.67%**↑ over the initial student, with a standard deviation of **0.66**. Scaling and transfer experiments show positive results for the smaller Qwen student and under Llama (Grattafiori et al., 2024) transfer. Controlled ablations cover answer access, filtering, loss, unlikelihood regularization, and repair search. A matched one-round efficiency test further quantifies the cost–performance trade-off: REPAIR-OPD uses $1.36\times$ OPD's end-to-end training time and has a seven-benchmark mean response length of $1.69\times$ OPD's. We make three contributions:

① ***A future-oriented view of OPD.*** We study how token-level teacher interventions reshape the student's autoregressive future and identify the induced continuation, rather than the corrected token alone, as the object to evaluate. This reframes OPD from teacher-supervision fitting on observed trajectories to outcome-verified repair of student-generated futures, opening a future-oriented direction for on-policy distillation.

② ***A new post-training method.*** To our knowledge, REPAIR-OPD is the first OPD method to combine local teacher replacement, student-generated full-suffix regeneration, and outcome-verified wrong-to-correct learning. It locates a repair point, lets the student regenerate from the corrected prefix, and learns only from **wrong-to-correct trajectories** whose futures pass outcome verification.

③ ***Empirical validation.*** Experiments across benchmarks, model scales, and transfer settings show consistent improvements over OPD baselines, supported by controlled ablations. As discussed above, REPAIR-OPD reaches **54.55%** overall in the main setting, a **4.72%**↑ over the strongest baseline. The efficiency analysis further shows that the added training time and response length constitute an acceptable cost–performance trade-off.

2 RELATED WORK

OPD trains a student with teacher supervision on trajectories sampled from its current policy, reducing the mismatch between training and autoregressive generation (Gu et al., 2024; Agarwal et al., 2024). Subsequent methods improve this paradigm along two technical lines.

(I) Supervision design. The first line improves OPD supervision through two complementary forms: **context-based methods** and **objective-based methods**. ❶ **Context-based methods.** OPSD (Zhao et al., 2026) uses privileged traces; OPCD (Ye et al., 2026) adds auxiliary context; MOPD (Yu et al., 2026c) uses peer successes and failures; SSOPD (Tan & Hong, 2026) distills from the shortest correct rollout into prefixes of the longest wrong rollout; and OPSDL (Zhang et al., 2026b) uses a short-context self-teacher. ❷ **Objective-based methods.** EOPD (Jin et al., 2026) uses entropy-aware Kullback–Leibler (KL) regularization; Veto (Jang et al., 2026) reformulates targets; SCOPE (Zheng et al., 2026) weights losses by correctness and perplexity; G-OPD and its ExOPD variant (Yang et al., 2026) adjust the reward–KL balance; Fu et al. (2026) match the teacher's top-K support; PACED (Xu et al., 2026a) weights problems by competence; REOPOLD (Ko et al., 2026) clips token rewards; vOPD (Oh et al., 2026) adds a control-variate baseline; and EGRSD (Ke et al., 2026) gates updates by uncertainty. Li et al. (2026) analyze teacher–student compatibility, while Yu et al. (2026a) characterize the geometry of OPD updates.

(II) Token selection. The second line selects token positions to optimize. Zhang et al. (2026a) focus OPD on reasoning prefixes; TSD-KD (Kim & Baek, 2026) selects student-uncertain but teacher-confident tokens; TIP (Xu et al., 2026b) characterizes uncertain and overconfident tokens; TA-OPD (Wang et al., 2026) keeps locally teachable disagreements; TrOPD (Xing et al., 2026)

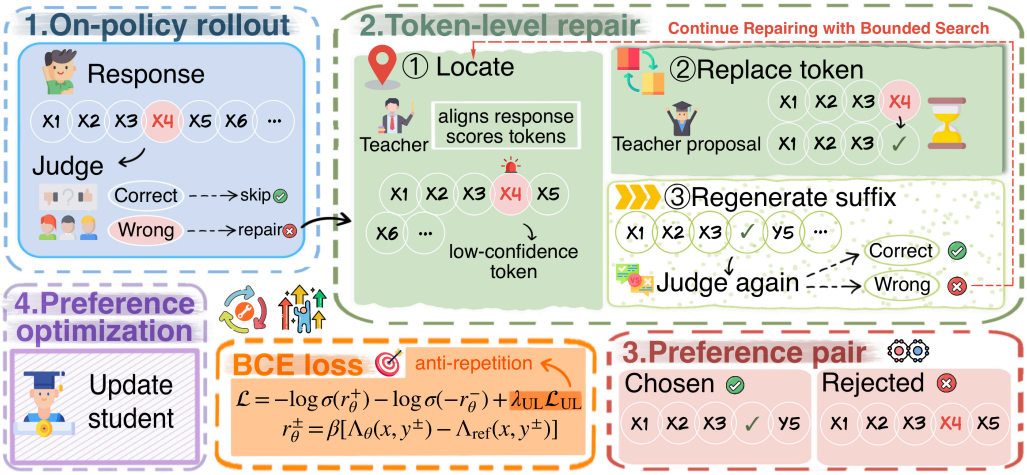


Figure 2: Overview of the four-stage REPAIR-OPD loop: on-policy rollout, token-level repair, preference-pair construction, and preference optimization, repeated cyclically.

builds agreement-based trust regions; and DEAR (Xiao et al., 2026) finds decision-supporting tokens. In a code-generation analysis, Lee et al. (2026) show that isolated peak-disagreement positions can sometimes redirect the continuation.

Despite these advances, existing OPD methods mainly improve supervision on observed student trajectories, without directly testing whether a local correction improves the student's **autoregressive future**. In contrast, REPAIR-OPD uses **disagreement** to locate a local correction and **repairs the future** by letting the student regenerate the complete continuation from the corrected prefix; only outcome-verified wrong-to-correct trajectories are retained for learning.

3 METHODOLOGY

Problem Setup. Let $\mathcal{D} = (x_k, a_k) * k = 1^N$ be a training set, $(x_k,\ a_k)$ denotes Q-A pair. Note a trainable student $p*\theta$, a frozen teacher p_T, a reference model p_{ref}, and a fixed outcome Judge J. The answer a is privileged input to the teacher and Judge. The student generates its response based on x. Only θ is optimized. For each problem, the current student samples an on-policy response $\mathbf{y}^{(0)}$. A completed response rejected by the Judge is passed to Repair, and a training pair is created only when Repair turns this original incorrect response $\mathbf{y}^-$ into a response $\mathbf{y}^+$ accepted by the Judge.

3.1 OVERVIEW

As shown in Figure 2, REPAIR-OPD turns local token feedback into outcome-verified trajectory repair. Given an incorrect on-policy rollout, it uses teacher–student disagreement to locate candidate intervention points, replaces one local decision with a teacher alternative, and lets the current student regenerate the remaining trajectory. The Judge retains only wrong-to-correct repairs, which are then used for preference learning.

3.2 REPAIR POINTS

For an incorrect trajectory $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{V}_S^n$, let $h_i^S = (x, \mathbf{y}_{<i})$ be the student context and let h_i^T be the corresponding teacher context, the reference final answer a, and the decoded student prefix. The teacher evaluates the observed student trajectory by teacher forcing. At each aligned position, we take the highest-probability returned teacher candidate whose text differs from the observed token. Its text is re-encoded with the student tokenizer as a nonempty span $\boldsymbol{\tau}_i = (\tau_{i,1}, \ldots, \tau_{i,m_i})$, where $m_i \geq 1$.

Given h_i^T, $q_T^0(i)$ is T's conditional probability for the aligned observed token u_i^0; $q_T^\star(i)$ is T's conditional probability for the proposed alternative $u_i^\star$. Their difference is $\Delta_T(i)$:

$$q_T^0(i) = p_T(u_i^0 \mid h_i^T), \qquad q_T^\star(i) = p_T(u_i^\star \mid h_i^T), \qquad \Delta_T(i) = q_T^\star(i) - q_T^0(i). \tag{1}$$

We screen teacher proposals with $q_T^\star(i) \geq \tau_q$ and $\Delta_T(i) \geq \tau_\Delta$; the threshold values and their rationale are given in Appendix A. We then apply a task-specific mathematical-text filter to the student-decoded replacement. The filter retains mathematical expressions, predefined function names, and eligible single-letter symbols, and is applied only to the replacement text.

The student scores its observed token and the replacement span under the same prefix:

$$q_S^0(i) = p_\theta(y_i \mid h_i^S), \qquad q_S^\star(i) = \exp\left(\frac{1}{m_i}\sum_{j=1}^{m_i} \log p_\theta(\tau_{i,j} \mid h_i^S, \boldsymbol{\tau}_{i,<j})\right). \tag{2}$$

Thus, a multi-token replacement is scored by its geometric mean token probability. With $[v]_+ = \max(0, v)$, we define

$$\Delta_S(i) = q_S^0(i) - q_S^\star(i), \qquad G_i = \Delta_T(i) + [\Delta_S(i)]_+. \tag{3}$$

We retain candidates with $G_i \geq \tau_G$ and rank them using *earliest-high-gap*: among high-gap candidates, earlier positions are preferred.

3.3 Repair and Learning

For a current search state $\mathbf{y}$ and a selected proposal $(i, \boldsymbol{\tau}_i)$, let $\oplus$ denote sequence concatenation. The repair transition is

$$\mathbf{z}_i \sim p_\theta(\cdot \mid x, \mathbf{y}_{<i} \oplus \boldsymbol{\tau}_i), \qquad \widetilde{\mathbf{y}}^{(i)} = \mathbf{y}_{<i} \oplus \boldsymbol{\tau}_i \oplus \mathbf{z}_i. \tag{4}$$

This discards y_i and the complete old suffix $\mathbf{y}_{\geq i}$. The teacher supplies only the local replacement, while the current student generates the complete new continuation.

Algorithm 1 formalizes the bounded repair procedure, producing Judge-verified wrong-to-correct preference pairs that provide the training signal for subsequent preference optimization.

Algorithm 1 Bounded Breadth-First Repair Search and Preference-Pair Construction

Input: Question x, answer a, rollout $\mathbf{y}^{(0)}$; student p_θ, teacher p_T, Judge J.
Limits: $\mathrm{D}_{\max}$ (interventions per path), $\mathrm{B}_{\max}$ (branches per state), and $\mathrm{A}_{\max}$ (total attempts).
Output: Verified pair $(x, \mathbf{y}^+, \mathbf{y}^-)$, or $\emptyset$.

if $\mathbf{y}^{(0)}$ is truncated or $J(x, a, \mathbf{y}^{(0)}) = 1$ **then return** $\emptyset$
$\mathcal{Q} \leftarrow [(\mathbf{y}^{(0)}, 1, 0)]$; $n \leftarrow 0$ ▷ FIFO queue; d counts applied interventions
while $\mathcal{Q} \neq \emptyset$ **and** $n < \mathrm{A}_{\max}$ **do**
  $(\mathbf{y}, s, d) \leftarrow \textsc{PopFront}(\mathcal{Q})$
  # Rank repair candidates on the current trajectory.
  $\mathcal{C} \leftarrow$ recompute and rank candidates $(i, \boldsymbol{\tau}_i)$ with $i \geq s$ using teacher–student gaps
  $\mathcal{C} \leftarrow \textsc{First}(\mathcal{C}, \min(\mathrm{B}_{\max}, \mathrm{A}_{\max} - n))$; $n \leftarrow n + |\mathcal{C}|$
  # Apply local teacher corrections and regenerate the complete suffix.
  For each $(i, \boldsymbol{\tau}_i) \in \mathcal{C}$, replace y_i and regenerate the complete suffix with p_θ
  # Return verified successes; queue eligible failures for further repair.
  for $(i, \boldsymbol{\tau}_i) \in \mathcal{C}$ in ranked order **do**
    if $\widetilde{\mathbf{y}}^{(i)}$ is truncated **then continue**
    if $J(x, a, \widetilde{\mathbf{y}}^{(i)}) = 1$ **then return** $(x, \widetilde{\mathbf{y}}^{(i)}, \mathbf{y}^{(0)})$
    if $d + 1 < \mathrm{D}_{\max}$ **and** $n < \mathrm{A}_{\max}$ **then**
      $\textsc{PushBack}(\mathcal{Q}, (\widetilde{\mathbf{y}}^{(i)}, i + |\boldsymbol{\tau}_i|, d + 1))$
return $\emptyset$

We optimize the verified pairs with BCE. For $M \in \{\theta, \mathrm{ref}\}$, define the summed response log-likelihood

$$\Lambda_M(x, \mathbf{y}) = \sum_{j=1}^{|\mathbf{y}|} \log p_M(y_j \mid x, \mathbf{y}_{<j}), \tag{5}$$

where p_θ is the student and p_{ref} is frozen. For a scale $\beta > 0$, the reference-relative scores are

$$r_\theta^+ = \beta\big(\Lambda_\theta(x, \mathbf{y}^+) - \Lambda_{\mathrm{ref}}(x, \mathbf{y}^+)\big), \qquad r_\theta^- = \beta\big(\Lambda_\theta(x, \mathbf{y}^-) - \Lambda_{\mathrm{ref}}(x, \mathbf{y}^-)\big). \tag{6}$$

For a preference batch $\mathcal{B}_{\text{pair}}$, with $\sigma(t) = (1 + e^{-t})^{-1}$, the loss is

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{B}_{\text{pair}}|} \sum_{(x,\mathbf{y}^+,\mathbf{y}^-)\in\mathcal{B}_{\text{pair}}} \left[\log \sigma(r_\theta^+) + \log \sigma(-r_\theta^-)\right]. \tag{7}$$

When an original rollout reaches the generation limit and contains an exact periodic suffix, the repetition gate bypasses Judge evaluation and Repair and produces a loop sample $b = (x_b, \mathbf{y}_b, \mathcal{N}_b)$, where $\mathcal{N}_b$ contains the marked repeated-token positions. For a loop batch $\mathcal{B}_{\text{loop}}$, the token-level unlikelihood (UL) loss and complete objective are

$$\begin{aligned} \mathcal{L}_{\text{UL}} &= -\frac{1}{|\mathcal{B}_{\text{loop}}|} \sum_{b\in\mathcal{B}_{\text{loop}}} \frac{1}{|\mathcal{N}_b|} \sum_{j\in\mathcal{N}_b} \log(1 - p_\theta(y_{b,j} \mid x_b, \mathbf{y}_{b,<j})), \\ \mathcal{L} &= \mathcal{L}_{\text{BCE}} + \lambda_{\text{UL}}\mathcal{L}_{\text{UL}}. \end{aligned} \tag{8}$$

We set $\mathcal{L}_{\text{UL}} = 0$ when no loop sample is available. Loop samples are incorporated only with a preference update and do not trigger a separate update.

4 Experiments

We evaluate whether RePair-OPD improves mathematical reasoning, transfers beyond the training domain, remains effective under different model and training configurations, and incurs manageable computational costs. Our experiments address the following research questions (RQs).

RQ1: How does RePair-OPD perform on in-domain and out-of-domain benchmarks?

RQ2: How robust is RePair-OPD across model scales, families, and random seeds?

RQ3: How do answer access, filtering, training, and repair settings affect performance?

RQ4: What computational overhead does RePair-OPD incur during training and inference?

4.1 Experimental Setup

Dataset. We train on 13,476 English samples from DAPO-Math-17k-Processed (Yu et al., 2026b). We evaluate on MATH-500 (Lightman et al., 2024), AIME 2024[1], AIME 2025[2], AIME 2026[3], AMC 2023[4], MMLU-Pro (Wang et al., 2024), and GPQA Diamond (Rein et al., 2023).

Evaluation. We report pass@8 on MATH-500, AIME 2024–2026, and AMC 2023, and exact-choice accuracy on MMLU-Pro and GPQA Diamond. Detailed decoding settings and shared controls are provided in Appendix B.

Configuration. We use `Qwen3-8B` as the teacher and `Qwen3-4B` (Yang et al., 2025) as the student, with a frozen copy of the student initialization as the reference model. The outcome Judge is `Qwen3.6-27B`[5]. The teacher, reference model, and Judge remain fixed throughout training.

Baselines. We compare RePair-OPD with the original OPD (Agarwal et al., 2024), and two groups of extensions: **(I) supervision design**, EOPD (Jin et al., 2026), Stable OPD (ATR) (Jang et al., 2026), and SCOPE (Zheng et al., 2026); and **(II) token selection**, TIP (Xu et al., 2026b) and TA-OPD (Wang et al., 2026).

4.2 Main Results

4.2.1 Main Comparison (RQ1)

To answer RQ1, we compare RePair-OPD with the student model and six OPD baselines using `Qwen3-8B` as the teacher and `Qwen3-4B` as the student. The RePair-OPD setting uses $(\mathrm{D}_{\max}, \mathrm{B}_{\max}, \mathrm{A}_{\max}) = (4, 3, 32)$.

[1] https://huggingface.co/datasets/HuggingFaceH4/aime_2024
[2] https://huggingface.co/datasets/opencompass/AIME2025
[3] https://huggingface.co/datasets/MathArena/aime_2026
[4] https://maa.org/math-competitions/amc-1012
[5] https://qwen.ai/blog?id=qwen3.6-27b

Table 1: Main comparison (%) with `Qwen3-8B` teacher and `Qwen3-4B` student; REPAIR-OPD is the three-seed mean. Arrows show changes from the student; bold entries mark best scores.

Method	In-domain Mathematics						Out-of-domain			Overall
	MATH-500	AIME-2024	AIME-2025	AIME-2026	AMC-2023	Mean	MMLU-Pro	GPQA-D	Mean	Mean
`Qwen3-4B`	86.80%	23.33%	20.00%	20.00%	73.91%	44.81%	59.46%	37.63%	48.54%	45.88%
`Qwen3-8B`	92.20%	43.33%	33.33%	26.67%	84.78%	56.06%	63.17%	39.39%	51.28%	54.70%
OPD	88.20% ↑1.40	36.67% ↑13.33	23.33% ↑3.33	23.33% ↑3.33	71.74% ↓2.17	48.65% ↑3.85	58.41% ↓1.05	39.90% ↑2.27	49.15% ↑0.61	48.80% ↑2.92
EOPD	**92.20%** ↑5.40	30.00% ↑6.67	26.67% ↑6.67	16.67% ↓3.33	71.74% ↓2.17	47.45% ↑2.65	59.18% ↓0.28	39.02% ↑1.39	49.10% ↑0.56	47.92% ↑2.05
TIP	87.40% ↑0.60	23.33%	16.67% ↓3.33	16.67% ↓3.33	71.74% ↓2.17	43.16% ↓1.65	58.63% ↓0.83	**41.29%** ↑3.66	**49.96%** ↑1.42	45.10% ↓0.77
TA-OPD	90.20% ↑3.40	33.33% ↑10.00	**30.00%** ↑10.00	26.67% ↑6.67	69.57% ↓4.35	49.95% ↑5.14	58.24% ↓1.22	40.78% ↑3.15	49.51% ↑0.97	49.83% ↑3.95
ATR	89.60% ↑2.80	26.67% ↑3.33	**30.00%** ↑10.00	20.00%	73.91%	48.04% ↑3.23	58.10% ↓1.36	39.77% ↑2.14	48.94% ↑0.40	48.29% ↑2.42
SCOPE	89.80% ↑3.00	30.00% ↑6.67	26.67% ↑6.67	20.00%	76.09% ↑2.17	48.51% ↑3.70	59.02% ↓0.44	39.14% ↑1.51	49.08% ↑0.54	48.67% ↑2.80
REPAIR-OPD	92.13% ↑5.33	**40.00%** ↑16.67	28.89% ↑8.89	**36.67%** ↑16.67	**85.51%** ↑11.59	**56.64%** ↑11.83	**60.59%** ↑1.13	38.09% ↑0.46	49.34% ↑0.80	**54.55%** ↑8.67

Obs. 1: REPAIR-OPD leads on the in-domain and overall averages. It reaches 56.64 in-domain versus 49.95 for TA-OPD, and 49.34 out-of-domain, close to TIP's best 49.96. Its overall mean is 54.55, exceeding TA-OPD, the strongest baseline overall at 49.83, by 4.72%↑.

4.2.2 MODEL SCALING AND TRANSFER (RQ2)

To answer RQ2, we compare with OPD as the basic OPD method and TA-OPD as the strongest baseline in Table 1, while including the undistilled student as a reference. We evaluate one smaller Qwen setting and one cross-family Llama (Grattafiori et al., 2024) setting.

Scaling Laws. We scale `Qwen3-4B` as the teacher and `Qwen3-0.6B` as the student, with $(\mathrm{D}_{\max}, \mathrm{B}_{\max}, \mathrm{A}_{\max}) = (5, 3, 48)$. We can also list the similar observations:

Table 2: Model scaling (%) with a `Qwen3-4B` teacher and a `Qwen3-0.6B` student. Arrows show changes from the student; bold entries mark best scores.

Method	In-domain Mathematics						Out-of-domain			Overall
	MATH-500	AIME-2024	AIME-2025	AIME-2026	AMC-2023	Mean	MMLU-Pro	GPQA-D	Mean	Mean
`Qwen3-0.6B`	60.40%	13.33%	0.00%	0.00%	26.09%	19.96%	27.24%	27.65%	27.45%	22.10%
`Qwen3-4B`	86.80%	23.33%	20.00%	20.00%	73.91%	44.81%	59.46%	37.63%	48.54%	45.88%
OPD	61.40% ↑1.00	0.00% ↓13.33	6.67% ↑6.67	0.00%	36.96% ↑10.87	21.01% ↑1.04	27.55% ↑0.31	23.86% ↓3.79	25.71% ↓1.74	22.35% ↑0.25
TA-OPD	**72.80%** ↑12.40	6.67% ↓6.66	3.33% ↑3.33	**3.33%** ↑3.33	30.43% ↑4.34	23.31% ↑3.35	**28.62%** ↑1.38	23.99% ↓3.66	26.31% ↓1.14	24.17% ↑2.07
REPAIR-OPD	71.00% ↑10.60	6.67% ↓6.66	**10.00%** ↑10.00	0.00%	**50.00%** ↑23.91	**27.53%** ↑7.57	28.51% ↑1.27	**27.90%** ↑0.25	**28.21%** ↑0.76	**27.73%** ↑5.62

Obs. 2: REPAIR-OPD remains strongest with the smaller student, but its advantage narrows. It achieves in-domain and out-of-domain means of 27.53 and 28.21, exceeding TA-OPD by 4.22%↑ and 1.90%↑, respectively. Its overall mean is 27.73, versus 22.35 for OPD and 24.17 for TA-OPD; the margin over TA-OPD changes 4.72 → 3.56 (1.16% ↓) relative to the main setting.

Cross-Family Transfer. We evaluate a `Llama-3.1-8B-Instruct` teacher and a `Llama-3.2-3B-Instruct` student with $(\mathrm{D}_{\max}, \mathrm{B}_{\max}, \mathrm{A}_{\max}) = (5, 4, 96)$.

Table 3: Cross-family transfer (%) with a `Llama-3.1-8B-Instruct` teacher and a `Llama-3.2-3B-Instruct` student. Arrows show changes from the student; bold denotes best.

Method	In-domain Mathematics						Out-of-domain			Overall
	MATH-500	AIME 2024	AIME 2025	AIME 2026	AMC 2023	Mean	MMLU-Pro	GPQA-D	Mean	Mean
`Llama-3.2-3B-Instruct`	64.40%	10.00%	0.00%	0.00%	34.78%	21.84%	29.71%	19.70%	24.71%	22.66%
`Llama-3.1-8B-Instruct`	70.20%	10.00%	0.00%	3.33%	34.78%	23.66%	43.32%	22.85%	33.09%	26.36%
OPD	54.80% ↓9.60	3.33% ↓6.67	0.00%	0.00%	32.61% ↓2.17	18.15% ↓3.69	22.93% ↓6.78	17.93% ↓1.77	20.43% ↓4.28	18.80% ↓3.86
TA-OPD	55.00% ↓9.40	0.00% ↓10.00	**3.33%** ↑3.33	0.00%	30.43% ↓4.35	17.75% ↓4.08	23.90% ↓5.81	13.26% ↓6.44	18.58% ↓6.12	17.99% ↓4.67
REPAIR-OPD	**67.20%** ↑2.80	**13.33%** ↑3.33	0.00%	**3.33%** ↑3.33	**39.13%** ↑4.35	**24.60%** ↑2.76	**29.88%** ↑0.17	17.68% ↓2.02	23.78% ↓0.93	**24.36%** ↑1.71

Obs. 3: REPAIR-OPD transfers positively across model families. Its overall mean is 24.36, compared with 18.80 for OPD and 17.99 for TA-OPD; the in-domain and out-of-domain means are 24.60 and 23.78, respectively. Comparing benchmark outputs, we find that OPD and TA-OPD exhibit the same bimodal degradation: short guesses or repetitive, truncated rollouts. This shared generation instability lowers the valid-solution rate and explains the transfer drop.

4.2.3 STABILITY ACROSS RANDOM SEEDS (RQ2)

To answer RQ2, we repeat REPAIR-OPD with various seeds using `Qwen3-8B` as the teacher and `Qwen3-4B` as the student, with $(D_{max}, B_{max}, A_{max}) = (4, 3, 32)$; all other settings are fixed across runs. For reference, Table 4 also includes TA-OPD, the best baseline by Overall Mean in Table 1.

Table 4: Three-seed performance (%) with `Qwen3-8B` teacher and `Qwen3-4B` student. Arrows compare matched seeds or means against TA-OPD; bold/underlined scores rank first/second.

Method	In-domain Mathematics						Out-of-domain			Overall
	MATH-500	AIME-2024	AIME-2025	AIME-2026	AMC-2023	Mean	MMLU-Pro	GPQA-D	Mean	Mean
TA-OPD (Seed 7)	87.60%	30.00%	26.67%	16.67%	73.91%	46.97%	58.41%	40.03%	49.22%	47.61%
TA-OPD (Seed 42)	90.20%	33.33%	30.00%	26.67%	69.57%	49.95%	58.24%	40.78%	49.51%	49.83%
TA-OPD (Seed 123)	88.20%	33.33%	30.00%	23.33%	71.74%	49.32%	58.60%	**42.17%**	**50.39%**	49.63%
TA-OPD (Mean)	88.67%	32.22%	28.89%	22.22%	71.74%	48.75%	58.42%	40.99%	49.71%	49.02%
REPAIR-OPD (Seed 7)	**92.80%** ↑5.20	33.33% ↑3.33	30.00% ↑3.33	**43.33%** ↑26.67	84.78% ↑10.87	56.85% ↑9.88	**60.65%** ↑2.24	38.38% ↓1.65	49.52% ↑0.30	54.76% ↑7.15
REPAIR-OPD (Seed 42)	91.20% ↑1.00	40.00% ↑6.67	23.33% ↓6.67	36.67% ↑10.00	84.78% ↑15.22	55.20% ↑5.25	60.60% ↑2.36	39.02% ↓1.76	49.81% ↑0.30	53.66% ↑3.83
REPAIR-OPD (Seed 123)	92.40% ↑4.20	**46.67%** ↑13.33	**33.33%** ↑3.33	30.00% ↑6.67	**86.96%** ↑15.22	**57.87%** ↑8.55	60.52% ↑1.92	36.87% ↓5.30	48.70% ↓1.69	**55.25%** ↑5.62
REPAIR-OPD (Mean)	92.13% ↑3.47	40.00% ↑7.78	28.89%	36.67% ↑14.44	85.51% ↑13.77	56.64% ↑7.89	60.59% ↑2.17	38.09% ↓2.90	49.34% ↓0.37	54.55% ↑5.53

Obs. 4: REPAIR-OPD delivers consistent overall gains across training seeds. Its overall scores span $53.66 \sim 55.25$, with a mean of 54.55 and a population standard deviation of 0.66. Against TA-OPD at the same seed, the overall gains span $3.83\% \sim 7.15\%\uparrow$, while the three-seed mean increases $49.02 \rightarrow 54.55$ ($5.53\%\uparrow$). The gains across all three matched-seed comparisons and the limited score variation support the consistency of the improvement across the three training seeds.

4.3 ABLATION STUDY (RQ3)

To answer RQ3, we conduct all ablations with `Qwen3-8B` as the teacher and `Qwen3-4B` as the student. Given limited GPU resources, all ablations use a fixed random seed unless specified. We tune components independently, retain the best setting, and carry it forward to a subsequent experiment.



Figure 3: Training ablations (%). Labels in (a)–(d) give Overall scores; (e) shows three seeds per UL weight, with means and ± 1 population standard deviation. Markers denote individual seeds, while bars and lines report the corresponding aggregate means.

Answer Access. Answer access controls whether the teacher receives the known correct final answer as private guidance when scoring the student trajectory. The results are shown in Figure 3(a).

Obs. 5: Teacher answer access improves performance in both domains. With answer access, the in-domain mean moves $52.09 \rightarrow 56.85$ ($4.76\%\uparrow$). The out-of-domain mean moves $37.42 \rightarrow 49.52$ ($12.10\%\uparrow$), a larger gain that highlights the value of answer-conditioned guidance during transfer.

Filtering Rules. The task-specific filter acts only on the teacher-proposed replacement after teacher-confidence screening, retaining mathematical expressions, predefined function names, and eligible single-letter symbols. We compare it with an unfiltered variant; results are shown in Figure 3(b).

Obs. 6: The mathematical-text filter improves both domains, increasing their means by 3.54 in-domain and 1.11 out-of-domain; the larger in-domain gain indicates that filtering most directly improves training-distribution repair quality while retaining a smaller transfer benefit.

Loss Function. We compare three objectives on repaired preference pairs. Supervised fine-tuning (SFT) applies token-level cross-entropy only to the repaired trajectory, BCE independently increases its reference-relative score and decreases that of the original incorrect trajectory, and direct preference optimization (DPO) optimizes their relative score margin. Figure 3(c) summarizes the results.

Obs. 7: BCE provides the strongest aggregate performance. It reaches an overall mean of 54.76, outperforming SFT and DPO by 2.02%↑ and 2.83%↑, respectively, and obtains the best in-domain mean of 56.85. SFT achieves the highest out-of-domain mean of 51.65. The 2.02% $\sim$ 2.83%↑ margins over both alternatives show that BCE provides the most consistent aggregate objective.

UL Regularization. The UL weight λ_{UL} scales the token-level UL penalty on repeated tokens to mitigate infinite loops; when enabled, we set it to 0.05, and Figure 3(d) shows the corresponding comparison across regularized and unregularized training settings in this ablation.

Obs. 8: UL regularization improves overall performance. Enabling UL moves the overall mean 50.00 $\rightarrow$ 54.76 (4.76%↑). Without UL, training occasionally enters unstable infinite-generation loops; by suppressing these failures during optimization, UL preserves the effective update budget and improves both training stability and final accuracy across runs.

UL Weight. After establishing the benefit of UL, we next examine how its weight affects performance and training stability. We compare $\lambda_{\mathrm{UL}} \in \{0.05, 0.10\}$ across three random seeds to assess robustness, and Figure 3(e) reports both individual runs and the corresponding three-seed means.

Obs. 9: A moderate UL weight is more reliable across seeds. With $\lambda_{\mathrm{UL}} = 0.05$, the overall mean and population standard deviation are 54.55 ± 0.66, compared with 53.09 ± 1.94 for $\lambda_{\mathrm{UL}} = 0.10$. Although the larger weight reaches 55.73 under one seed, its lower results under the other seeds make it less consistent. Its population deviation is nearly 3× larger (1.94 vs. 0.66), indicating that a moderate penalty better balances performance and training stability.

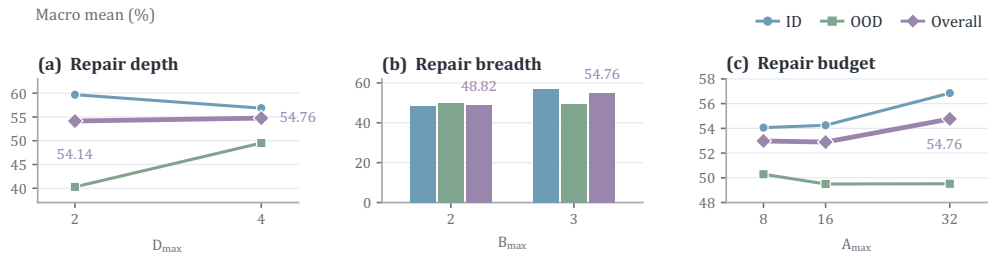


Figure 4: Repair-search ablations (%); panel (b) uses grouped bars. Panels vary repair depth, breadth, and budget, respectively, and report how each search limit changes in-domain, out-of-domain, and overall macro means under the same training setup.

Repair Configuration. We next examine the repair-search configuration by varying depth, breadth, and budget independently; these controls limit intervention steps, candidates per state, and total continuations, respectively, with results summarized below.

Repair Depth $\mathbf{D_{max}}$. $\mathrm{D}_{\max}$ limits successive interventions along a single repair trajectory: smaller depths favor shallow, localized corrections, whereas larger depths permit longer repair chains and broader search. We compare $\mathrm{D}_{\max} = 2$ and $\mathrm{D}_{\max} = 4$; Figure 4(a) reports the results.

Obs. 10: $\mathrm{D}_{\max}$ has a limited effect on aggregate performance, and successful repairs are generally shallow. Increasing $\mathrm{D}_{\max}$ $2 \rightarrow 4$ changes the overall mean by only 0.62%↑. After a useful local intervention, the student typically requires few further corrections to regenerate a viable continuation, leaving most effective repair paths short and avoiding unnecessary search expansion.

Repair Breadth $\mathbf{B_{max}}$. $\mathrm{B}_{\max}$ limits ranked candidate interventions expanded from each state: smaller breadth focuses on top candidates, whereas larger breadth explores more alternatives. We compare $\mathrm{B}_{\max} = 2$ and $\mathrm{B}_{\max} = 3$; Figure 4(b) reports the results.

Obs. 11: A larger $\mathrm{B}_{\max}$ improves performance, and successful repairs often depend on finding a suitable intervention point. The overall mean moves $48.82 \rightarrow 54.76$ ($5.94\%\uparrow$) when $\mathrm{B}_{\max}$ increases $2 \rightarrow 3$, indicating the value of broader candidate coverage over extending one branch. The $5.94\%\uparrow$ gain shows that broader candidate coverage is more valuable than extending a single search branch, because it exposes suitable corrections earlier in the bounded search.

Repair Budget $\mathbf{A_{max}}$. $\mathrm{A}_{\max}$ caps repaired continuations evaluated during search: smaller budgets limit computation, whereas larger budgets verify more alternatives. We compare $\mathrm{A}_{\max} = 8$, $\mathrm{A}_{\max} = 16$, and $\mathrm{A}_{\max} = 32$; Figure 4(c) reports the results.

Obs. 12: Increasing $\mathrm{A}_{\max}$ can improve performance, but the benefit is limited because most errors are repaired within a few attempts. The overall means for $\mathrm{A}_{\max} = 8$, 16, and 32 are 52.98, 52.89, and 54.76, respectively. Nearly all of the improvement appears at $\mathrm{A}_{\max} = 32$, which recovers harder cases missed by smaller budgets, while the near-flat $8 \rightarrow 16$ change indicates diminishing returns before the largest budget, confirming that most corrections need only modest search.

4.4 Efficiency Analysis (RQ4)

To answer RQ4, we compare the training and inference costs of REPAIR-OPD and OPD with a `Qwen3-8B` teacher and a `Qwen3-4B` student; Figure 5 summarizes the results, while detailed measurements and analyses are provided in Appendix C.4.

Training Efficiency. We measure end-to-end time for one training round on 1,000 matched source questions with the same prompts, precision, and sequence limits.

Obs. 13: REPAIR-OPD raises training time modestly while delivering a substantial performance gain. REPAIR-OPD takes 5,376.3 seconds versus 3,960.8 for OPD ($1.36\times$), while its overall mean of 54.55 exceeds the strongest baseline by $4.72\%\uparrow$. The limited time increase relative to this accuracy improvement keeps the added training cost controllable.

Inference Efficiency. We compare mean response lengths excluding prompts across seven benchmarks, with REPAIR-OPD averaged over three seeds and the base student included as a reference.

Obs. 14: REPAIR-OPD produces longer inference responses while improving performance. Its seven-benchmark macro mean reaches 1,365 tokens, compared with 808.30 for OPD ($1.69\times$). Although this extra generation length increases inference cost, the accompanying accuracy improvement makes the trade-off acceptable for the evaluated benchmarks.

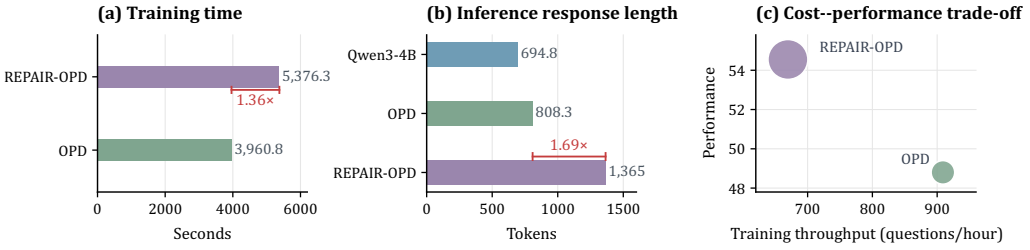


Figure 5: Efficiency comparison on matched questions and seven-benchmark macro means. Panels (a)–(b) show training time and response length; panel (c) relates training throughput to Overall performance, with bubble radius encoding response length. Red labels mark relative multipliers.

5 Conclusion

Past errors may already have occurred, but the future they lead to can still be rewritten. We propose **REPAIR-OPD**, a bounded breadth-first algorithm where teacher–student disagreement locates repair points, local corrections in turn regenerate futures, and Judge verification converts changed outcomes into preference pairs. Local interventions thus redirect futures while bounded recovery stays practical; accordingly, broader evaluation and efficient search are next directions.

AI USE STATEMENT

Generative artificial intelligence (AI) tools were used to assist with manuscript writing and language polishing, as well as with the development of research code. AI tools were also used to analyze and organize related literature. They additionally provided inspiration for figure concepts and visual presentation. The figures themselves were drawn manually by the authors. The authors reviewed and revised all AI-assisted text and code and manually checked the reported experimental results.

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.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

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.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting self-uncertainty in on-policy self-distillation for efficient LLM reasoning. *arXiv preprint arXiv:2605.13255*, 2026.

Minsang Kim and Seung Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations*, volume 2026, pp. 103174–103196, 2026.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Taeckyung Lee, Jeonghye Kim, Hyungjun Yoon, and Sung-Ju Lee. Dense supervision but sparse guidance: A token-level analysis of on-policy self-distillation. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 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.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. KL for a KL: On-policy distillation with control variate baseline. *arXiv preprint arXiv:2605.07865*, 2026.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Zhiquan Tan and Yinrong Hong. Self-supervised on-policy distillation for reasoning language models. *arXiv preprint arXiv:2605.17497*, 2026.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026.

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, 2024.

Jinwei Xiao, Zhuowen Han, Yueqing Sun, Zhengxi Lu, Yuxin Liu, Zhiyuan Yao, Wentao Chen, Qi Gu, and Xunliang Cai. Finding the evidence: Discovering decision-supporting tokens for on-policy reasoning distillation. *arXiv preprint arXiv:2606.22830*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026a.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026b.

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.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense supervision, sparse updates: On the sparsity and geometry of on-policy distillation. *arXiv preprint arXiv:2606.13657*, 2026a.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026b.

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

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E Tillman. Fast and effective on-policy distillation from reasoning prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, 2026a.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. OPSDL: On-policy self-distillation for long-context language models. *arXiv preprint arXiv:2604.17535*, 2026b.

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.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

A Threshold Selection

We use three fixed thresholds for repair-candidate screening and ranking. The symbols used in the main text are set as follows:

$$\tau_q = 0.15, \qquad \tau_\Delta = 0.20, \qquad \tau_G = 0.25. \tag{9}$$

Here τ_q is the confidence floor for the proposed teacher alternative, τ_Δ is the required teacher margin over the aligned observed token, and τ_G is the minimum combined teacher–student score for retaining a candidate. We selected these values through extensive preliminary experiments on the training set, evaluating multiple threshold configurations. This setting consistently provided a favorable balance between repair success and search efficiency, while maintaining stable training and downstream performance. Because of limited GPU resources, we did not conduct an exhaustive sensitivity study over all alternatives.

B Experimental Setup Details

B.1 Teacher Input Construction

We use the following teacher input template:

User Use the known correct final answer as private guidance when continuing the solution.

Problem:
Write a short solution with only the essential steps. Stop after the answer.
[Problem]

Known correct final answer:
[Reference final answer]

Assistant *[Current student trajectory]*

B.2 Evaluation

For MATH-500, AIME 2024–2026, and AMC 2023, we sample eight responses per question with temperature 0.8, top-$p = 0.95$, and a maximum generation length of 8192 tokens. We report pass@8, counting a question as solved when at least one response is mathematically equivalent to the reference answer. For MMLU-Pro and GPQA Diamond, we use greedy decoding with temperature 0, top-$p = 1.0$, the same generation limit, and report exact-choice accuracy under 5-shot and zero-shot chain-of-thought prompting, respectively. Responses whose answers cannot be extracted by the applicable rule are counted as incorrect.

B.3 Prompting and Teacher Controls

We use the prompt prefix **"Write a short solution with only the essential steps. Stop after the answer."** for all methods during training and open-ended mathematical evaluation. Thinking mode is disabled whenever available, and every teacher receives the known correct final answer as private guidance. The exact template appears in Appendix B.1.

B.4 Training Environment and Details

We conduct all experiments on one machine with eight 80 GB NVIDIA A800 GPUs, using PyTorch 2.11.0 (Paszke et al., 2019) and CUDA 13.0. Four GPUs host the Judge, one hosts the teacher, and three train the student. We use verl (Sheng et al., 2025) as the training framework and vLLM (Kwon et al., 2023) version 0.22.0 for model serving and rollout generation. Both REPAIR-OPD and all baselines are trained for one on-policy round using full-parameter fine-tuning. We use AdamW, a constant learning rate of 5×10^{-6}, a preference batch size of 8, and a maximum training sequence length of 8192. Training rollouts use temperature 0.6 and top-$p = 0.95$.

C Analysis

This appendix analyzes the one-round main configuration with a `Qwen3-8B` teacher and a `Qwen3-4B` student. Unless stated otherwise, the analysis uses the three runs with seeds 7, 42, and 123, $\mathrm{D}_{\max} = 4$, $\mathrm{B}_{\max} = 3$, and $\mathrm{A}_{\max} = 32$.

C.1 Training Curves

Figure 6 separates three aspects of training: preference learning, original-answer accuracy, and the recovery of initially incorrect responses. Each column shows one random seed.

Reward dynamics. Chosen rewards rise and rejected rewards decrease overall, widening the reward margin in all three runs. For seed 7, the mean margin increases from $0.208 \rightarrow 0.825$ over the first and last 15 updates ($0.617\%\uparrow$). This growing separation indicates that the student increasingly favors verified repairs over the original incorrect responses relative to the frozen reference model.

Original accuracy. Strict accuracy measures how often the student's response is judged correct before repair. All three runs show higher accuracy in the final training quarter than in the first; for seed 7, it moves $30.07\% \rightarrow 36.60\%$ ($6.53\%\uparrow$). The rolling curves fluctuate across the shuffled training problems, with a broad upward trend rather than monotonic improvement.

Effective repair. The effective repair rate is the fraction of initially incorrect responses successfully corrected by Repair. It fluctuates throughout training, with modestly higher rates in the final quarter across all three seeds. Over each complete run, Repair recovers $31.47\% \sim 31.96\%$ of initial errors, providing verified wrong-to-correct pairs even as original-answer accuracy improves.

C.2 Training Data Statistics

Each seed processes the same 13,476 non-empty examples from the filtered DAPO-Math input, so the three runs contain 40,428 rollout presentations but only 13,476 distinct questions. A successful Repair creates one preference pair for the main BCE objective, whereas a detected repetition creates a separate UL sample. The resulting event counts are shown in Figure 7.

The optimizer consumed 8,481 preference pairs and 367 UL samples, for 8,848 update-driving samples in total, or 21.9% of the 40,428 rollout presentations; 19 pairs remained in the asynchronous buffer at the end of the runs. Thus fewer than one quarter of the rollouts directly drove parameter updates, yet this selectively routed subset supplied the learning signal for the gains reported in the main experiments. The remaining slots are not useless: many are already correct, some remain uncorrected under the bounded search, and repetition cases use the auxiliary UL path.

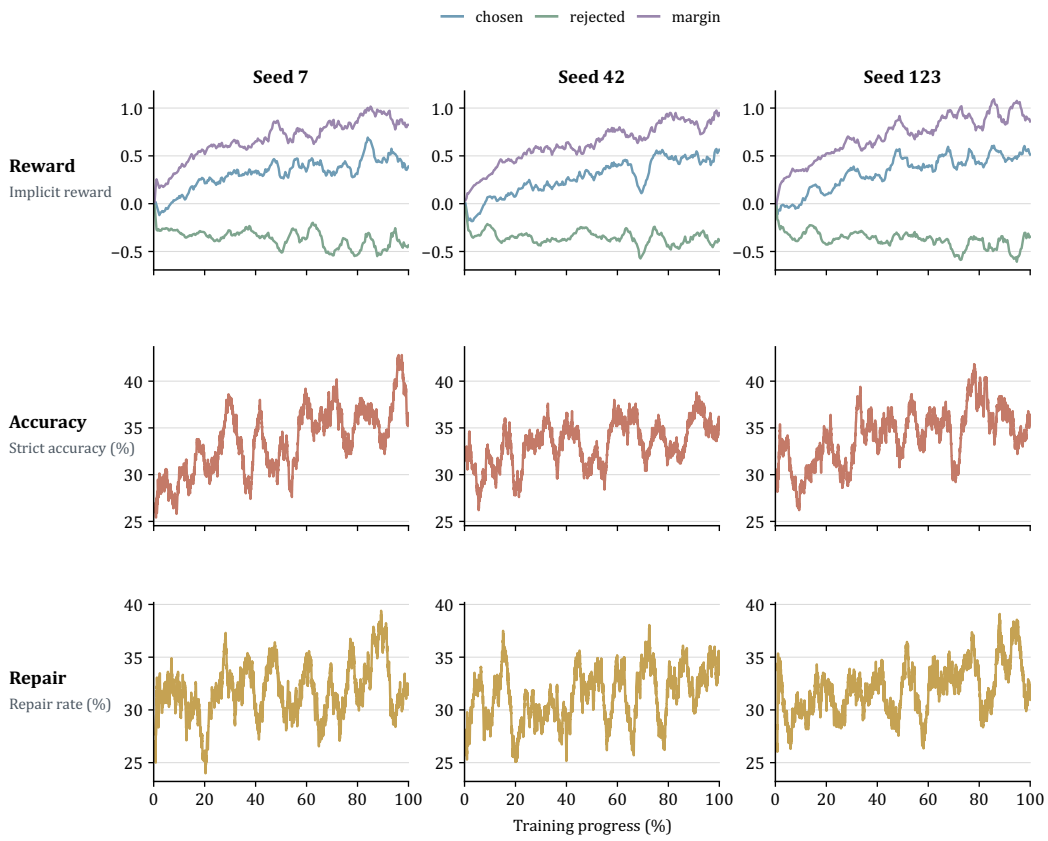


Figure 6: Training-time reward and on-policy outcome statistics. Rows show reward dynamics, original accuracy, and effective repair; columns separate seeds 7, 42, and 123. Curves use trailing windows of 15 updates or 500 rollout slots.

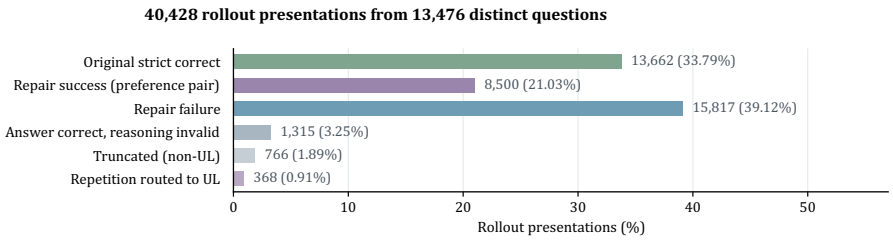


Figure 7: Training-event counts and rollout fractions over three seeds.

C.3 Repair Mechanism

C.3.1 Repair Position

We analyze the selected intervention position for the 8,500 successful pairs. The absolute position has mean 316.53 and median 210.5 tokens; the largest bins are positions 129–256 (21.55%) and 257–512 (23.22%). With relative position defined as $p = (i + 1)/L$, the mean and median are 45.15% and 38.03%. The first, middle, and last quarters contain 37.60%, 37.21%, and 25.19% of successful interventions, indicating that successful repairs are more common before the final quarter. The complete distribution is visualized in Figure 8(a).

C.3.2 Repair Depth, Breadth, and Attempts

Repair search is controlled by three budgets. $\mathrm{D}_{\max}$ bounds the number of successive interventions along one search path, $\mathrm{B}_{\max}$ bounds the number of highest-ranked candidates expanded from each visited state, and $\mathrm{A}_{\max}$ caps the total number of candidate continuations launched by the search. For example, with $\mathrm{D}_{\max} = 3$ and $\mathrm{B}_{\max} = 2$, the first three levels can launch at most $2 + 4 + 8 = 14$ candidates; setting $\mathrm{A}_{\max} = 6$ stops the search after six launches. We report the realized depth D as the number of interventions on a successful path, so $D = 1$ is a one-step repair and $D = 4$ reaches the configured depth limit. The realized attempt count A includes every candidate launched in a batch, including candidates generated in parallel with the one that succeeds.

Among 8,500 successful pairs, depth is concentrated near the root: 54.18% finish at $D = 1$, 23.54% at $D = 2$, 15.09% at $D = 3$, and 7.19% at $D = 4$, with mean depth 1.753. Attempts are similarly low: the median is 3, the mean is 7.01, and 77.98% of successful pairs use at most nine candidate continuations; only 0.66% reach $A = 32$. Breadth is directly observable for the 4,605 $D = 1$ cases, where the root is the only visited state: 95.77% expand all three ranked candidates. These distributions show that successful repairs usually require a shallow path and limited search, while retaining a small tail of cases that benefit from broader exploration.

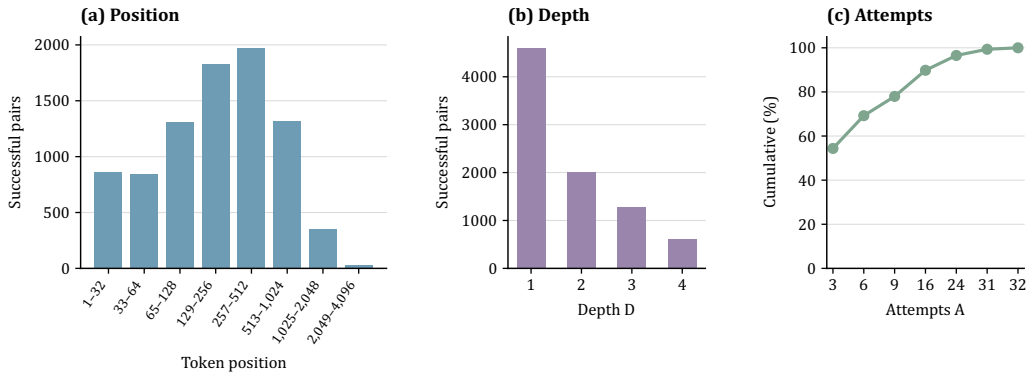


Figure 8: Repair position, depth, and candidate-attempt distributions for successful pairs.

Figure 8(b) shows the depth distribution, and panel (c) ends at the configured $\mathrm{A}_{\max} = 32$.

C.4 Efficiency Analysis

C.4.1 Training Efficiency

We compare end-to-end processing of 1,000 matched source questions using the same models, prompt prefix, precision, sequence limits, and one-round setting. The OPD arm uses the exact full-vocabulary forward-KL objective; the REPAIR-OPD arm includes rollout, teacher calls, Repair, Judge verification, and updates. REPAIR-OPD takes 5,376.3 seconds versus 3,960.8 seconds for OPD, or 1.357 times the training time. Throughput is 669.6 versus 908.9 source questions per hour. The extra search and verification add 35.7%↑ training time, while the main-setting overall mean is 4.72% higher than the strongest OPD baseline. This cost-performance trade-off is favorable: the bounded repair search converts additional training computation into a clear accuracy gain.

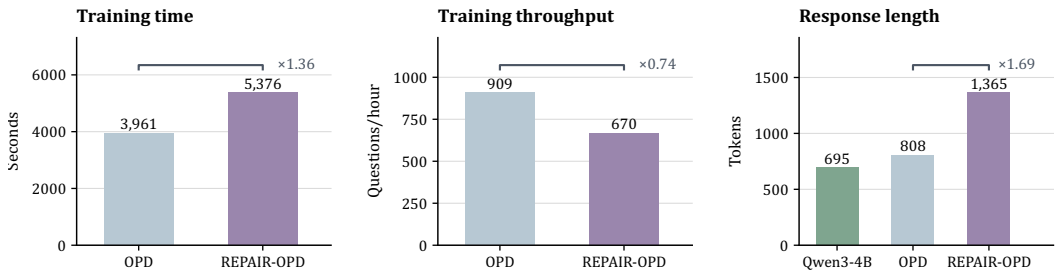


Figure 9: Training time, training throughput, and response length. Comparison lines show the ratios between the paired bars.

C.4.2 Inference Efficiency

Table 5 reports mean response lengths excluding prompts, with REPAIR-OPD averaged over three seeds. Its seven-benchmark macro mean is 1,365 tokens, compared with 694.77 for the initial `Qwen3-4B` student and 808.30 for OPD, or $1.69\times$ the OPD length (Figure 9). Responses are longer on all seven benchmarks, with the largest absolute increases over OPD on AIME 2024 and GPQA Diamond. These additional output tokens represent an inference cost accompanying the accuracy gains in Table 1. Thus, REPAIR-OPD trades a measurable increase in generation length for substantially stronger accuracy.

Table 5: Mean response length in tokens, excluding prompts. REPAIR-OPD uses the three-seed mean with a `Qwen3-8B` teacher and a `Qwen3-4B` student.

Benchmark	`Qwen3-4B`	OPD	REPAIR-OPD
MATH-500	294	445	507
AIME 2024	966	1005	1979
AIME 2025	701	945	1698
AIME 2026	961	1178	1836
AMC 2023	467	710	949
MMLU-Pro	343	348	597
GPQA Diamond	1130	1027	1991
Macro mean	694.77	808.30	1365.00

D Case Study

We report two complete successful repairs from the main-results configuration. Each case includes the original problem input, the full response before Repair, the exact token replacement, the full response after each regeneration, and the Judge outcome.

D.1 Case 1: One-Step Repair

Input.

Prefix: Write a short solution with only the essential steps. Stop after the answer.
Problem. AB is the diameter of a circle centered at O. C is a point on the circle such that angle $\angle BOC$ is 60°. If the diameter of the circle is 5 inches, find the length of chord AC, expressed in inches. The answer should be in the form $\frac{k\sqrt{m}}{n}$, where k, m, and n are integers. Please calculate the value of $k+m+n$.
Known final answer: 10.

Original student response.

We are given:
- AB is the diameter of a circle with center O, so $AB = 5$ inches.
- C is a point on the circle such that $\angle BOC = 60°$.
- We are to find the length of chord AC.

Step 1: Use the Law of Cosines in triangle AOC.
Since AB is the diameter, A and B are endpoints of the diameter, so O is the midpoint of AB, and $OA = OB = \frac{5}{2} = 2.5$.
Also, since C is on the circle, $OC = OA = OB = 2.5$.
So triangle AOC has sides $OA = 2.5$ and $OC = 2.5$, and the student incorrectly sets $\angle AOC = 60°$ because $\angle BOC = 60°$. The student then applies the Law of Cosines:

$$AC^2 = OA^2 + OC^2 - 2 \cdot OA \cdot OC \cos(60°),$$

$$AC^2 = (2.5)^2 + (2.5)^2 - 2 \cdot 2.5 \cdot 2.5 \cdot \frac{1}{2} = 6.25,$$

so $AC = \sqrt{6.25} = \frac{5}{2}$.
Step 2: Express in the form $\frac{k\sqrt{m}}{n}$. The response writes $AC = \frac{5}{2} = \frac{5\sqrt{1}}{2}$, so $k = 5$, $m = 1$, and $n = 2$.
Final answer: $k + m + n = 5 + 1 + 2 = \boxed{8}$.

Repair signal and regenerated response. The teacher proposes the replacement at response token 197:

$$6 \longrightarrow 1.$$

The student discards the old suffix and regenerates it from the patched prefix:

We are given:
- AB is the diameter of a circle with center O, so $AB = 5$ inches.
- C is a point on the circle such that $\angle BOC = 60°$.
- We are to find the length of chord AC.

Step 1: Use the Law of Cosines in triangle AOC.
Since AB is the diameter, A and B are endpoints of the diameter, so O is the midpoint of AB, and $OA = OB = \frac{5}{2} = 2.5$. Also, $OC = 2.5$. The corrected angle is $\angle AOC = \angle AOB - \angle BOC = 180° - 60° = 120°$.
Applying the Law of Cosines gives

$$AC^2 = (2.5)^2 + (2.5)^2 - 2(2.5)(2.5)\cos(120°) = 18.75,$$

$$AC = \sqrt{18.75} = \sqrt{\frac{75}{4}} = \frac{5\sqrt{3}}{2}.$$

Therefore $k = 5$, $m = 3$, and $n = 2$, so

$$k + m + n = 5 + 3 + 2 = \boxed{10}.$$

The Judge accepts this regenerated response.

D.2 Case 2: Two-Step Repair

Input.

Prefix: Write a short solution with only the essential steps. Stop after the answer.
Problem. The point $P = (1, 2, 3)$ is reflected in the xy-plane, then its image Q is rotated by $180°$ about the x-axis to produce R, and finally, R is translated by 5 units in the positive-y direction to produce S. What are the coordinates of S? Provide your answer in the form (x, y, z), and calculate the value of $x + y + z$.
Known final answer: $S = (1, 3, 3)$ and $x + y + z = 7$.

Original student response.

1. Reflect $P = (1, 2, 3)$ in the xy-plane:

$$Q = (1, 2, -3).$$

2. Rotate Q by 180° about the x-axis:

$$R = (-1, -2, -3).$$

3. Translate R by 5 units in the positive-y direction:

$$S = (-1, 7, -3).$$

Final answer: $S = (-1, 7, -3)$ and $x + y + z = -1 + 7 - 3 = \boxed{3}$.

Repair step 1 (depth 1). The teacher proposes the following replacement:

$$\text{response index} = 68, \qquad (- \longrightarrow (.$$

The complete response after suffix regeneration is:

1. Reflect $P = (1, 2, 3)$ in the xy-plane:

$$Q = (1, 2, -3).$$

2. Rotate Q by 180° about the x-axis:

$$R = (1, -2, 3).$$

3. Translate R by 5 units in the positive-y direction:

$$S = (1, 7, 3).$$

Final answer: $S = (1, 7, 3)$ and $x + y + z = 1 + 7 + 3 = \boxed{11}$.

The Judge still marks this response as wrong.

Repair step 2 (depth 2). The next candidate is selected from the newly generated response:

$$\text{response index} = 106, \qquad 7 \longrightarrow 3.$$

The complete final response after the second replacement and suffix regeneration is:

1. Reflect $P = (1, 2, 3)$ in the xy-plane:

$$Q = (1, 2, -3).$$

2. Rotate Q by 180° about the x-axis:

$$R = (1, -2, 3).$$

3. Translate R by 5 units in the positive-y direction:

$$S = (1, 3, 3).$$

Final answer: $S = (1, 3, 3)$ and $x + y + z = 1 + 3 + 3 = \boxed{7}$.

The Judge accepts the final response.