LEARNING FROM LOCAL REPAIRS: EDIT-CERTIFIED ADVANTAGE REDISTRIBUTION FOR RUBRIC-BASED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubric-based reinforcement learning provides richer feedback than binary rewards, but still broadcasts one response-level advantage across all generated tokens. On-policy self-distillation (OPSD) instead derives token-level guidance from distribution shifts induced by privileged context. However, privileged context can also alter wording and style, making distribution shifts an ambiguous basis for token-level credit assignment. To address this ambiguity, we introduce ECAR (Edit-Certified Advantage Redistribution), which uses successful local revisions to obtain response-specific evidence for token-level credit. The failed criteria guide a reviser toward a local repair that preserves unrelated content. A revision is certified when it remains local, yields a correct answer, improves the weighted rubric score, and preserves previously satisfied criteria. Aligned edits and their neighborhoods define edit-certified repair support on the original response. ECAR uses this support to redistribute the original advantage while preserving its response-level mean and each token advantage's sign. Responses without such support retain the standard sequence-level RL update. ECAR achieves higher average performance than reinforcement learning and self-distillation baselines across four mathematical reasoning test benchmarks. Ablations further establish the importance of both the revision certificate and edit-based localization. These results highlight the value of certified local repairs for token-level credit assignment.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become a central approach to improving mathematical reasoning in language models (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025; Cui et al., 2026). An automatic verifier supplies a scalable outcome signal, but outcome and reasoning operate at different resolutions. A single reward must credit a response that may contain useful intermediate steps, one decisive error, and many tokens that merely express the solution. Process verifiers and rubric-based reinforcement learning enrich this signal with step-level judgments or query-specific criteria (Lightman et al., 2024; Wang et al., 2024; Gunjal et al., 2026; Xu et al., 2026). ROPD, for example, induces a rubric from teacher references and current-policy rollouts, verifies each rollout against the rubric, and optimizes the resulting score on policy (Fang et al., 2026). Yet its group-relative advantage remains a response-level quantity broadcast uniformly across all sampled tokens. A richer outcome can distinguish better responses without revealing which tokens made them better.

On-policy self-distillation (OPSD) appears to solve this granularity problem. The current policy is given privileged context, such as a verified solution, and the induced token-distribution shift supervises the same policy without that context (Zhao et al., 2026; Hübotter et al., 2026). The resulting signal is dense, but its meaning is ambiguous. Privileged context can alter the mathematical path, yet it can also shorten a derivation, change notation, remove hesitation, or increase confidence globally. RLCSD contrasts positive and negative privileged contexts to cancel shared stylistic drift (Pan et al., 2026). SRPO routes samples between RL and self-distillation (Li et al., 2026a), while RLSD uses outcome advantages to set update directions and self-teacher signals to modulate their magnitudes (Yang et al., 2026). These mechanisms regulate how teacher signals affect learning, but token-level guidance still derives from distribution changes under privileged context. The unresolved

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

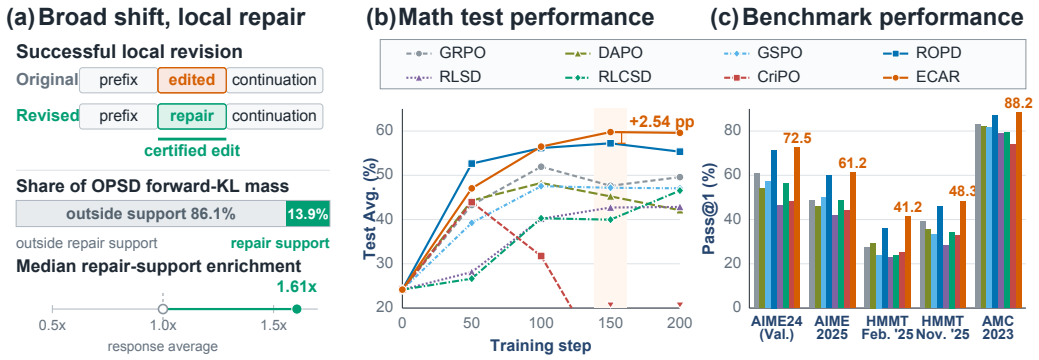

Figure 1: **ECAR overview.** (a) **Motivation.** OPSD's mean per-token forward KL within repair support is only 1.61× the response mean (median across responses), indicating weak enrichment around successful local repairs. (b) **Test performance dynamics.** ECAR attains the highest Test Avg. by step 150 and retains the lead at step 200. (c) **Performance across benchmarks.** ECAR outperforms the RL baselines on all four mathematical test benchmarks; AIME24 is shown separately as validation performance. Results in (b,c) use Qwen3-4B.

question comes one step earlier: *does that change identify a repairable error in this particular sampled response?*

Our analysis shows that OPSD probability shifts are weakly aligned with successful local repairs. For each response, we identify the region implicated by a successful local revision and measure how OPSD's token-level forward-KL signal is distributed inside and outside it. Across responses, the median per-token signal within the repair support is only 1.61× the response average, while tokens outside the support account for 86.07% of the total forward-KL mass. The OPSD signal is therefore broadly distributed beyond the region implicated by the repair.

This observation motivates using successful local revisions to identify where changes can improve the current response. The current policy is prompted to make the smallest local correction needed to satisfy the criteria that the response fails, while preserving unrelated content. We accept a revision only when it improves the response without regressing previously satisfied criteria. Aligned edits and their neighborhoods then define edit-certified repair support on the original response.

Building on this repair evidence, we introduce ECAR (Edit-Certified Advantage Redistribution). ECAR separates the direction, location, and token-level strength of credit assignment. The rubric-derived sequence advantage sets the sign and baseline magnitude of token advantages. Edit-certified repair support identifies where local weighting applies. Within this region, ECAR uses each original token's probability drop from the standard context to the revision context to determine its local weight. These weights guide the redistribution of the sequence advantage across tokens in the original on-policy response. If a response has no edit-certified repair support, all its tokens retain the original sequence advantage.

Experiments on mathematical reasoning benchmarks show that ECAR outperforms outcome-level RL, pure OPSD, and RL–OPSD baselines, with consistent improvements across three Qwen3 model scales.

Our contributions are:

- **We show that OPSD probability shifts are weakly aligned with successful local repairs.** By aligning OPSD probability shifts with repair support from successful local revisions on the same rollouts, we find that probability change is only weakly enriched around repaired regions and remains broadly distributed beyond them. It therefore provides limited localization of these observed repairs.
- **We introduce a revision certificate that turns successful local revisions into response-specific evidence for adjusting token-level credit.** A revision is certified only when it produces a correct answer, improves the weighted rubric score, repairs at least one failed criterion, preserves all previously satisfied criteria, and remains local. Aligned edits and

their neighborhoods define *edit-certified repair support* on the original rollout, providing a response-specific region for repair-based credit redistribution.

- **We introduce Edit-Certified Advantage Redistribution for token-level credit assignment.** ECAR combines the rubric-derived sequence advantage, edit-certified repair support, and the one-sided probability drop of original sampled actions under the revision context. It constructs a zero-sum token correction that redistributes the sequence advantage over the original response while preserving the mean advantage and each token advantage's sign. Together, these signals turn self-distillation into a mechanism for allocating outcome credit within the original on-policy response.
- **We demonstrate the effectiveness of ECAR.** ECAR outperforms outcome-level RL, pure OPSD, and RL–OPSD baselines on mathematical reasoning benchmarks, with consistent gains across three Qwen3 model scales. Controlled ablations and signal-localization analyses show that revision certification, edit localization, and localized redistribution each contribute to the gains.

2 Related Work

Outcome, process, and rubric supervision. Outcome-reward RL includes GRPO (Shao et al., 2024), DAPO's dynamic sampling and asymmetric clipping (Yu et al., 2025), and GSPO's sequence-level importance ratios (Zheng et al., 2025). Process supervision instead evaluates intermediate reasoning (Lightman et al., 2024; Wang et al., 2024; Cui et al., 2026). Rubric-based RL assesses responses against query-specific criteria (Gunjal et al., 2026). ROPD derives these criteria from teacher references and current-policy rollouts, then aggregates verification results into a response-level reward (Fang et al., 2026). The resulting sequence advantage is still shared by all tokens. To provide token-level supervision, Rubrics-to-Tokens learns which tokens are relevant to each criterion (Xu et al., 2026). ECAR instead uses successful local revisions to identify where to refine the rubric-derived advantage.

On-policy distillation and its combination with RL. Distillation trains a student using teacher distributions or teacher-generated sequences (Hinton et al., 2015; Kim & Rush, 2016). On-policy distillation (OPD) applies this supervision to student-generated responses, as in GKD and MiniLLM (Agarwal et al., 2024; Gu et al., 2024). OPSD obtains a self-teacher by giving the current policy privileged context (Zhao et al., 2026). Recent methods combine teacher guidance with outcome feedback through different mechanisms. SDPO uses feedback to construct the self-teacher (Hübotter et al., 2026), while SRPO and H^2SD select training objectives according to response correctness (Li et al., 2026a; Cai et al., 2026). RLSD complements outcome-based reinforcement learning with token-level guidance from a self-teacher (Yang et al., 2026). Distilled RL incorporates teacher guidance into the RL objective (Wang et al., 2026a), and Uni-OPD uses outcome rewards to improve its reliability (Hou et al., 2026). To improve the specificity of distillation signals, RLCSD contrasts correct-hint and wrong-hint contexts to reduce shared stylistic effects (Pan et al., 2026), while CREDIT separates input-specific guidance from patterns shared across inputs (Shen et al., 2026a).

Across these approaches, effective learning depends on teacher construction and objective design (Li et al., 2026b; Zhu et al., 2026; Fu et al., 2026; Oh et al., 2026). For credit assignment, probability changes under privileged context need not identify tokens that contribute to task success (Nguyen et al., 2026). Such guidance can suppress useful uncertainty and self-correction (Kim et al., 2026b; Kaur et al., 2026) or favor reference-specific shortcuts (Shen et al., 2026b). ECAR therefore uses successful local repairs as evidence for where token credit should be adjusted.

Revision-guided and localized learning. Self-Refine improves responses through iterative feedback (Madaan et al., 2023). HeRL uses failed trajectories and unmet rubrics as hindsight guidance for generating improved responses (Zhang et al., 2026). SD-Zero distills a reward-conditioned reviser into a generation policy (He et al., 2026). For localized supervision, TRACE focuses learning on annotated critical or erroneous spans (Wang et al., 2026b). Rubric-conditioned self-distillation builds self-teachers from structured criteria (Gu et al., 2026), and CriPO uses criterion-conditioned probability shifts to promote unexplored or suppressed behaviors (Xia et al., 2026). In ECAR, certified edits on the original on-policy response guide the redistribution of rubric-based outcome credit.

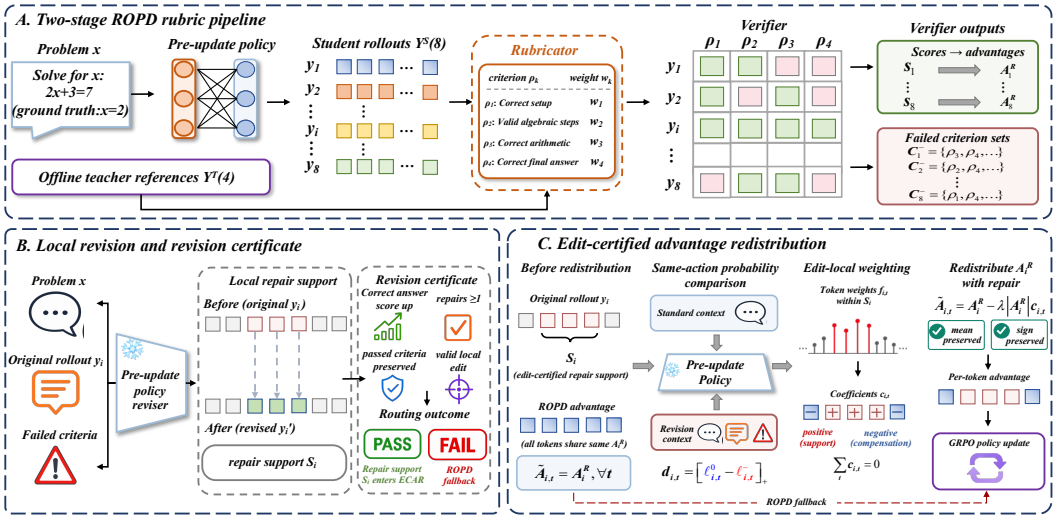

Figure 2: **ECAR training pipeline.** ROPD supplies a rubric-derived sequence advantage and failed criteria. A local revision is checked for improvement and locality, and certified edits define repair support on the original response. Probability drops within this support determine local weights for redistributing the ROPD advantage.

3 Method

3.1 Problem Formulation and Method Overview

ECAR redistributes the sequence advantage from rubric-based RL across tokens in the original sampled response. As shown in Figure 2, it uses certified local revisions to identify where credit should be adjusted, then assigns local token weights to redistribute the advantage.

Let π_{θ^-} denote the frozen pre-update policy. For a problem x, it samples G responses $y_i = (y_{i,1}, \ldots, y_{i,T_i})$. Following ROPD (Fang et al., 2026), a rubricator uses x, offline teacher references, and these rollouts to construct $\mathcal{C}(x) = \{c_k = (\rho_k, w_k)\}_{k=1}^{K_x}$, where ρ_k is a criterion and w_k its importance weight, with $\sum_k w_k > 0$. The verifier returns $v_{i,k} \in \{0, 1\}$ indicating whether y_i satisfies ρ_k. The weighted rubric score and centered sequence advantage are

$$s_i = \frac{\sum_{k=1}^{K_x} w_k v_{i,k}}{\sum_{k=1}^{K_x} w_k}, \qquad A_i^{\mathrm{R}} = s_i - \frac{1}{G}\sum_{j=1}^{G} s_j. \tag{1}$$

ROPD broadcasts A_i^{R} uniformly over valid response tokens in the standard clipped GRPO objective, without standard-deviation rescaling (Schulman et al., 2017; Shao et al., 2024). We call this the *ROPD update*.

Building on the ROPD update, ECAR first generates and certifies local revisions, then uses the certified edits to guide advantage redistribution over the original response.

3.2 Edit-Certified Repair Support

Failed rubric criteria guide the search for a local correction. Let $\mathcal{C}_i^-$ contain the criteria that y_i fails. A response is eligible for revision when $A_i^{\mathrm{R}} \neq 0$ and $1 \leq |\mathcal{C}_i^-| \leq K$, where K is the maximum number of failed criteria allowed for revision. The frozen pre-update policy receives the problem x, original response y_i, and failed criteria $\mathcal{C}_i^-$. It is prompted to make the smallest sufficient correction while preserving unrelated content. The revised response is sampled as $y_i' \sim \pi_{\theta^-}(\cdot \mid \mathcal{P}_{\mathrm{rev}}(x, y_i, \mathcal{C}_i^-))$, where $\mathcal{P}_{\mathrm{rev}}$ denotes the revision prompt instantiated from the template in Appendix B. We evaluate y_i' against the same query-specific rubric, obtaining criterion outcomes $v_{i,k}'$ and the revised score s_i'.

We validate the revision before using its edits as evidence for credit assignment. The acceptance certificate is

$$g_i = \mathbb{I}\big[\,\mathrm{AnsOK}(y_i') \wedge s_i' > s_i \wedge \textstyle\sum_{k:v_{i,k}=0} v_{i,k}' \geq 1 \\ \wedge \textstyle\sum_{k:v_{i,k}=1}(1 - v_{i,k}') = 0 \wedge \mathrm{GenValid}(y_i') \wedge \mathrm{Local}(y_i, y_i')\big]. \tag{2}$$

Here AnsOK compares the extracted final answer with the ground-truth answer under the task's answer-equivalence rule. The rubric conditions require a higher score, repair of at least one failed criterion, and preservation of all previously satisfied criteria. $\mathrm{GenValid}$ excludes truncated revisions and leaked private feedback. Local requires both character similarity and original-line retention to be at least a shared threshold τ, set to 0.7 by default; operational definitions are given in Appendix B.3.

For a certified pair (y_i, y_i'), deterministic token-ID alignment maps edits back to the original rollout. The edit core E_i contains replaced or deleted original positions and the nearest original anchor for each inserted span. We expand this core with a weighted neighborhood:

$$m_{i,t} = \begin{cases} 1, & t \in E_i, \\ \omega, & 0 < d(t, E_i) \leq r, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $d(t, E_i)$ is the distance from token t to the nearest edited position. We use $r = 8$ and $\omega = 0.25$. The nonzero positions $S_i = \{t : m_{i,t} > 0\}$ form the *edit-certified repair support*, covering actual edits and their local context. Edits may occur at multiple disjoint locations, including downstream corrections necessitated by an earlier repair. The support therefore follows the observed edits rather than automatically extending from the first error to the end of the response. Because unchanged neighboring tokens may also be included, inclusion in the support does not imply that a token is erroneous. Appendix C.2 provides an illustrative example.

3.3 Matched Probability Drop within Repair Support

Certified edits identify where local weighting applies. Within this region, ECAR refines token weights by measuring how much the revision context lowers the probability of each original token. We therefore evaluate the same original response under the standard context and the revision prompt $\mathcal{P}_{\mathrm{rev}}$. Both evaluations use the frozen pre-update parameters $\bar{\theta} = \mathrm{sg}(\theta^-)$, with gradients stopped:

$$\ell_{i,t}^0 = \log \pi_{\bar{\theta}}(y_{i,t} \mid x, y_{i,<t}), \\ \ell_{i,t}^- = \log \pi_{\bar{\theta}}\big(y_{i,t} \mid \mathcal{P}_{\mathrm{rev}}(x, y_i, \mathcal{C}_i^-), y_{i,<t}\big). \tag{4}$$

The original token prefix is identical in both teacher-forced evaluations. Only the conditioning prompt changes: the standard context contains the problem x, while the revision prompt also supplies the original response and its failed criteria. The revised response y_i' is used only to construct the repair-support mask $m_{i,t}$; it is not supplied as a target or privileged answer in these evaluations.

We measure how strongly the revision context disfavors each sampled action through its one-sided probability drop:

$$d_{i,t} = \big[\ell_{i,t}^0 - \ell_{i,t}^-\big]_+, \qquad u_{i,t} = 1 - \exp(-d_{i,t}). \tag{5}$$

Here $d_{i,t} > 0$ means that the revision context makes the sampled action less likely. We refer to $u_{i,t} \in [0, 1)$ as the *revision drop*, a bounded measure of this decrease. If the revision context instead makes the action more likely, ECAR sets the revision drop to zero. This score records decreases only; zero does not imply that the token is correct.

Edit evidence supplies a baseline local weight even when the probability drop is zero. The revision drop increases this weight, giving

$$f_{i,t} = m_{i,t}\big[\eta + (1 - \eta)u_{i,t}\big], \tag{6}$$

where $\eta = 0.5$ controls the baseline contribution. Every supported token receives at least $\eta m_{i,t}$ from the certified edit evidence, and larger revision drops increase its weight. These fixed weights redistribute the rubric-derived sequence advantage in the next stage.

3.4 Conservative Advantage Redistribution and Policy Update

Redistribution reduces positive token advantages, or strengthens negative ones, within the repair support. An equal compensation outside the support preserves the response-level mean advantage. Let U_i contain the valid token positions outside S_i. We apply redistribution only when the revision certificate passes, $A_i^{\mathrm{R}} \neq 0$, and both S_i and U_i are nonempty.

To obtain this behavior, we construct a correction that will be subtracted from the original advantage. Its values are positive within S_i and negative outside, with the compensation distributed uniformly over U_i:

$$\hat{c}_{i,t} = \begin{cases} f_{i,t}, & t \in S_i, \\ -\dfrac{\sum_{k \in S_i} f_{i,k}}{|U_i|}, & t \in U_i. \end{cases} \tag{7}$$

By construction, $\sum_t \hat{c}_{i,t} = 0$. Uniform compensation requires no further token ranking outside the repair support. To bound the largest correction while preserving this balance, we scale the entire correction vector by a common factor:

$$c_{i,t} = \frac{\hat{c}_{i,t}}{\max(1, \|\hat{\mathbf{c}}_i\|_\infty)}. \tag{8}$$

The resulting credit redistributes, rather than replaces, the original rubric-derived sequence advantage:

$$\widetilde{A}_{i,t} = A_i^{\mathrm{R}} - \lambda |A_i^{\mathrm{R}}| c_{i,t}, \qquad \lambda = 0.25. \tag{9}$$

The subtraction lowers advantages within the repair support and raises them outside, producing the redistribution described above. Since $\sum_t c_{i,t} = 0$ and $|c_{i,t}| \leq 1$, redistribution preserves the response-level mean advantage and, for $\lambda < 1$, the sign of every nonzero advantage. We replace the uniformly broadcast A_i^{R} in the standard clipped GRPO objective with $\widetilde{A}_{i,t}$. Policy ratios, clipping, token-mean aggregation, and optimization settings remain identical to ROPD.

If a response is ineligible for revision, its revision fails certification, or either token set is empty, we use $c_{i,t} = 0$ and $\widetilde{A}_{i,t} = A_i^{\mathrm{R}}$, recovering the original ROPD update for that response. Setting $\lambda = 0$ bypasses revision and the failed-criterion probability evaluation entirely, making the complete training path identical to ROPD.

4 Experiments

4.1 Experimental Setup

Data and model roles. We train Qwen3-4B in non-thinking mode on DAPO-Math-17K (Yu et al., 2025). GPT-5.4 generates four offline references per problem (*RawRef*). *CleanRef* retains problems with at least one correct reference, allowing one additional four-response generation round when the initial four are all incorrect. We keep all four responses from the first qualifying round for each retained problem, yielding 16,356 training problems. GPT-5.5 constructs rubrics and verifies criterion satisfaction. Final-answer correctness is checked by answer-equivalence rules. The frozen pre-update policy generates revisions.

Training protocol. All RL methods use the same training set, eight rollouts per problem, and a 200-update budget. Each update uses 32 problems; DAPO dynamically samples additional groups to fill this batch. We select checkpoints by AIME24 validation Pass@1 within this budget and repeat the comparison with Qwen3-1.7B and Qwen3-8B. Appendices F.2 and F.3 additionally report fixed step-200 comparisons and three-seed ECAR–ROPD results, showing that ECAR's gains persist at a common training endpoint and across seeds.

Baselines and evaluation. We compare outcome-reward RL methods GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025); rubric-based ROPD (Fang et al., 2026); pure self-distillation methods OPSD (Zhao et al., 2026) and u-OPSD (Li et al., 2026c); and RL–OPSD methods RLSD (Yang et al., 2026), RLCSD (Pan et al., 2026), and CriPO (Xia et al., 2026). For OPSD, we use the official OpenThoughts Math30K training recipe; for u-OPSD, we evaluate the

Table 1: **Main results with Qwen3-4B in non-thinking mode.** The GPT-5.4 teacher is shown as a reference. Bold marks column-best scores excluding this reference. Initial denotes the starting checkpoint. Scores are percentages; AIME24 is the validation benchmark, Test Avg. averages the four math test benchmarks, and IFEval is reported separately.

Method	AIME24 (val)	AIME25	HMMT-F	HMMT-N	AMC23	Test Avg.	IFEval
GPT-5.4 teacher	74.17	62.08	38.13	60.42	87.35	61.99	91.62
Qwen3-4B (initial)	21.67	18.33	10.42	6.46	61.45	24.16	80.14
GRPO	60.63	48.54	27.50	39.17	83.13	49.59	80.42
DAPO	54.17	45.83	29.38	35.62	82.23	48.27	80.06
GSPO	57.29	49.79	23.96	33.33	81.63	47.18	**81.13**
ROPD	71.46	59.79	36.04	46.04	87.05	57.23	80.56
OPSD	20.00	18.13	10.21	11.04	60.84	25.05	80.83
u-OPSD	25.21	22.08	12.08	11.46	68.67	28.57	80.83
RLSD	46.46	41.67	22.71	28.13	78.92	42.85	80.09
RLCSD	56.25	48.54	23.75	34.38	79.52	46.55	80.70
CriPO	48.33	43.96	25.00	32.71	74.10	43.94	80.07
ECAR (ours)	**72.50**	**61.25**	**41.25**	**48.33**	**88.25**	**59.77**	80.97

released Qwen3-4B checkpoint. Among the shared-budget RL methods, RLSD uses a correct-answer context to modulate an outcome-derived advantage, RLCSD contrasts correct and incorrect sibling contexts before applying the residual signal, and CriPO uses criterion-conditioned probability shifts to assign token-level credit. Appendix A.2 summarizes the principal baseline configurations. We evaluate Pass@1 on AIME 2024/2025, HMMT February/November 2025, and AMC 2023 with a 32K response limit. AIME and HMMT use 16 sampled responses per problem, while AMC23 uses four; each score is the mean correctness over these samples. We report AIME24 as validation performance and the macro average of AIME25, HMMT February/November 2025, and AMC23 as *Test Avg.* IFEval is reported separately using its official prompt-level strict evaluator. Full prompt templates and optimization settings are provided in Appendices B and A, respectively.

4.2 Main Results

ECAR reaches a Test Avg. of 59.77, exceeding ROPD, the strongest RL baseline, by 2.54 percentage points and GRPO by 10.19 (Table 1). It outperforms ROPD on all four tests, most strongly on HMMT-February, with a question-paired bootstrap 95% confidence interval of $[0.23, 4.95]$ for the Test Avg. gain (Figure 5). At step 200, ECAR still leads on all four tests (59.59 versus 55.34; Table 7). Across three independent step-200 runs, its mean gain is 3.39 points, positive in every run (Table 8).

Results across model scales. ECAR obtains the strongest Test Avg. at every scale (Tables 1 and 2), improving over ROPD by 1.95, 2.54, and 1.77 points at 1.7B, 4B, and 8B, respectively. It also exceeds GRPO, DAPO, and GSPO at every scale, extending the comparison across outcome-reward policy optimization methods. ECAR maintains comparable instruction-following performance on IFEval.

Distillation-based baselines. OPSD and u-OPSD improve the 4B Test Avg. over the initial checkpoint by 0.89 and 4.41 points, while RLSD, RLCSD, and CriPO remain below GRPO. RLSD's lower performance than GRPO agrees with a separate study of non-thinking Qwen3 models on DAPO-Math (Kim et al., 2026a). CriPO varies with model scale, falling below GRPO at 1.7B and 4B but surpassing it at 8B (55.24 versus 52.87). The analysis in Section 4.4 helps interpret these results: OPSD and RLCSD distribute probability-shift signals broadly beyond repair support and place more signal mass on style than task tokens. These observations motivate ECAR's use of successful local repairs to anchor token weighting to regions where revision improves the response.

4.3 Component Ablations

Table 3 first isolates reference quality: CleanRef improves ROPD by 0.97 points over RawRef. The remaining ablations test certification, edit locations, token weights, and the update formulation.

Table 2: **Cross-scale results in non-thinking mode.**

Model	Method	AIME24 (val)	AIME25	HMMT-F	HMMT-N	AMC23	Test Avg.	IFEval
Qwen3-1.7B	Initial	10.42	6.88	2.92	3.75	38.25	12.95	68.23
	GRPO	37.92	27.08	17.08	16.46	64.16	31.20	68.02
	DAPO	32.71	23.75	15.83	13.75	59.34	28.17	68.01
	GSPO	32.71	28.54	15.62	18.33	63.55	31.51	67.79
	ROPD	**45.42**	33.96	20.00	24.79	68.07	36.71	67.29
	RLSD	31.25	24.17	15.00	11.25	54.82	26.31	67.57
	RLCSD	27.71	20.63	14.79	12.92	58.43	26.69	67.39
	CriPO	4.17	3.13	0.63	3.13	40.36	11.81	68.97
	ECAR	45.21	**37.92**	**21.25**	**26.46**	**68.98**	**38.65**	**69.09**
Qwen3-8B	Initial	25.83	17.92	8.96	6.67	63.55	24.27	83.25
	GRPO	64.79	50.42	34.79	43.12	83.13	52.87	83.19
	DAPO	62.29	46.67	29.17	39.79	82.23	49.46	82.83
	GSPO	64.17	48.33	29.58	41.46	83.43	50.70	83.28
	ROPD	70.83	63.54	41.04	50.00	88.25	60.71	83.42
	RLSD	62.71	44.38	28.54	36.67	83.43	48.25	82.62
	RLCSD	58.54	45.42	27.08	40.42	81.93	48.71	**83.45**
	CriPO	69.17	55.63	36.67	45.83	82.83	55.24	83.16
	ECAR	**72.50**	**65.21**	**42.08**	**53.75**	**88.86**	**62.47**	83.13

Table 3: **Component ablations.** Δ is Test Avg. minus full ECAR's, in percentage points.

Variant	Configuration	AIME24 (val)	AIME25	HMMT Feb.	HMMT Nov.	AMC23	Test Avg.	Δ
ROPD (RawRef)	Unfiltered reference pools	68.54	58.96	35.83	45.00	85.24	56.26	-3.51
ROPD (CleanRef)	Correct-reference filter; one retry	71.46	59.79	36.04	46.04	87.05	57.23	-2.54
Joint local JSD	ROPD + local JSD	69.38	62.71	38.96	45.00	86.75	58.35	-1.42
w/o certificate	Any usable aligned edit	67.92	59.38	37.92	46.25	85.84	57.35	-2.43
Dense ungated	Full-response centering	64.17	56.04	34.58	42.71	84.94	54.57	-5.20
Random support	Equal-size random span	69.79	**63.75**	35.42	45.42	85.84	57.61	-2.16
Support-averaged drop	Mean drop; same mask and mass	67.08	60.83	38.75	**48.75**	83.73	58.02	-1.75
ECAR	Full method	**72.50**	61.25	**41.25**	48.33	**88.25**	**59.77**	–

Without certification, usable aligned edits yield 57.35 versus ECAR's 59.77. Random support retains certification but moves the edit-mask weights to an equal-size random contiguous span, yielding 57.61. Dense ungated centers revision drops over the full response without certification or localization, yielding 54.57. Both certification and edit alignment contribute to the gain.

Support-averaged drop replaces individual drops with their support-weighted mean, retaining certification, $m_{i,t}$, redistribution mass, and λ. Its 58.02 Test Avg. is 1.75 points below ECAR, supporting tokenwise strength beyond edit-core and neighborhood weighting at fixed locations and mass.

Joint local JSD adds a Jensen–Shannon loss to ROPD between standard- and revision-context distributions on the same certified support. ECAR instead redistributes the rubric-derived advantage, raising Test Avg. from 58.35 to 59.77 (+1.42 points). Local guidance is more effective here as credit redistribution than as an auxiliary distillation loss. Appendix C.3 provides the variant definitions.

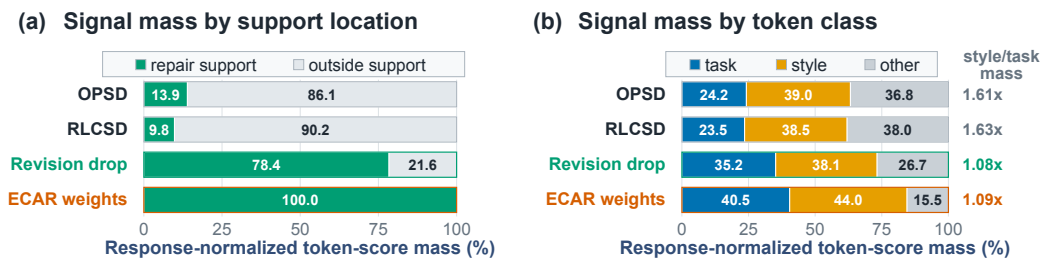

Figure 3: **Localization and task focus of token-level signals.** (a) Signal mass inside and outside repair support. ECAR shows the edit-local weights $f_{i,t}$, restricted to the support by construction. (b) Signal mass by RLCSD lexical token class (Pan et al., 2026). All signals use the same frozen Qwen3-4B before method-specific training; shares are normalized per response and then averaged.

Table 4: **Certificate selectivity.** K caps failed criteria for revision; τ sets the minimum character similarity and original-line retention. Other certificate conditions are unchanged.

Setting	K	τ	AIME24 (val)	AIME25	HMMT-F	HMMT-N	AMC23	Test Avg.	IFEval
No K, τ limits	∞	0.0	**72.50**	55.63	38.75	45.00	86.45	56.46	80.27
Relaxed K, τ	5	0.3	70.21	**64.58**	35.21	46.25	87.35	58.35	80.58
Full ECAR	3	0.7	**72.50**	61.25	**41.25**	**48.33**	**88.25**	**59.77**	**80.97**

4.4 Localization and Task Focus of Token-Level Signals

Figure 3 compares OPSD, RLCSD, revision drop $u_{i,t}$, and ECAR's edit-local weights $f_{i,t}$ on the same frozen Qwen3-4B before method-specific training.

Figure 3(a) averages response-normalized signal mass within repair support. OPSD's forward KL and RLCSD's absolute token log-probability contrast assign only 13.93% and 9.79% of their mass to the support, versus 7.52% for uniform token weighting. Revision drop raises this share to 78.39%. ECAR's edit-local weights $f_{i,t}$ are zero outside the support by construction; the final advantage correction also includes uniform compensation outside it. Appendix E.1 gives signal definitions and normalization details.

Figure 3(b) groups signal mass by RLCSD's lexical classes (Pan et al., 2026): *task* (mathematical expressions and terms), *style* (stylistic markers and function words), and *other* (remaining tokens; Appendix E.2). OPSD and RLCSD yield style-to-task ratios of $1.61\times$ and $1.63\times$, respectively. Revision drop reduces this ratio to $1.08\times$, with 35.20% task mass. ECAR's edit-local weights place 40.46% on task tokens and 44.03% on style tokens, a ratio of $1.09\times$. Revision conditioning and edit-local weighting thus bring task and style contributions closer to balance while focusing the signal on repair support.

4.5 How Selective Should the Certificate Be?

In Table 4, full ECAR exceeds the relaxed setting by 1.42 points and the setting without K, τ limits by 3.32 points. The relaxed setting performs best on AIME25 but has a lower mean: broader revision eligibility and weaker locality requirements do not consistently improve performance.

5 Conclusion

We introduced ECAR, which uses certified local revisions to obtain response-specific repair evidence. Matched probability drops redistribute the rubric-derived advantage within the original response while preserving its mean and each token advantage's sign. Successful repairs thus connect response-level rubric feedback to response-specific token-level learning. Across three Qwen3 scales, ECAR improves mathematical reasoning, with ablations supporting the complementary roles of certification, edit localization, and tokenwise strength. The current evidence concerns mathematical reasoning with reliable rubric verification and local revision; extending it to broader domains remains future work.

AI USE STATEMENT

Generative AI tools assisted with literature retrieval and discovery, manuscript language editing, code debugging, and organizing and packaging the supplementary code. We also used GPT models to generate teacher reference responses for training and to serve as model judges. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The method formulation and training procedure are described in Section 3. The appendix provides ECAR-specific configuration and prompts, certificate definitions, evaluation settings, and additional analyses. Shared rubricator and verifier templates follow ROPD (Fang et al., 2026). Code for the main experiments is included in the supplementary material.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Qiye Cai, Yichuan Ma, Peiji Li, Yongkang Chen, Qipeng Guo, Yicheng Zou, Linyang Li, Xiaocheng Feng, and Bing Qin. H^2SD: Hybrid hindsight self-distillation. *arXiv preprint arXiv:2607.18955*, 2026. URL https://arxiv.org/abs/2607.18955v4.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=9SkkifLopZ.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation. *arXiv preprint arXiv:2605.07396*, 2026. URL https://arxiv.org/abs/2605.07396.

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. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2603.25562.

Siyi Gu, Jialin Chen, Sophia Zhou, Arman Cohan, and Rex Ying. Rethinking reward supervision: Rubric-conditioned self-distillation. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2606.19327.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.17746.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026. URL https://arxiv.org/abs/2604.12002.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-OPD: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026. URL https://arxiv.org/abs/2605.03677.

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. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.20802.

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. URL https://arxiv.org/abs/2607.05184.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled RLVR. *arXiv preprint arXiv:2605.10781*, 2026a. URL https://arxiv.org/abs/2605.10781.

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? In *Conference on Language Modeling*, 2026b. URL https://arxiv.org/abs/2603.24472.

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. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. In *Conference on Language Modeling*, 2026a. URL https://arxiv.org/abs/2604.02288.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b. URL https://arxiv.org/abs/2604.13016.

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

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Xuan-Phi Nguyen, Zeyu Leo Liu, Yang Li, Shrey Pandit, Yiran Zhao, Anurag Koul, and Shafiq Joty. Privileged likelihood is not automatically value: Three checks for token credit in on-policy self-distillation. *arXiv preprint arXiv:2608.09263*, 2026. URL https://arxiv.org/abs/2608.09263.

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. URL https://arxiv.org/abs/2605.07865.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. RLCSD: Reinforcement learning with contrastive on-policy self-distillation. *arXiv preprint arXiv:2606.11709*, 2026. URL https://arxiv.org/abs/2606.11709.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, Gang Chen, and Jieping Ye. Purified OPSD: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026b. URL https://arxiv.org/abs/2607.02234.

Chen Wang, Zhaochun Li, Jionghao Bai, Yining Zhang, Hexuan Deng, Ge Lan, and Yue Wang. Distilled reinforcement learning for LLM post-training. *arXiv preprint arXiv:2607.17247*, 2026a. URL https://arxiv.org/abs/2607.17247.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. TRACE: Distilling where it matters via token-routed self on-policy alignment. *arXiv preprint arXiv:2605.10194*, 2026b. URL https://arxiv.org/abs/2605.10194.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9426–9439, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Mingxuan Xia, Yuhang Yang, Chao Ye, Shuai Zhu, Shenzhi Yang, Guangcheng Zhu, Yuhang Zhang, Cheng Peng, Haobo Wang, and Siqing Wang. CriPO: Enhancing rubric-based RL via self-distillation. *arXiv preprint arXiv:2607.18082*, 2026. URL https://arxiv.org/abs/2607.18082.

Tianze Xu, Yanzhao Zheng, Pengrui Lu, Lyumanshan Ye, Yong Wu, Zhentao Zhang, Yuanqiang Yu, Chao Ma, Jihuai Zhu, Pengfei Liu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, and Gang Yu. Rubrics to tokens: Bridging response-level rubrics and token-level rewards in instruction following tasks. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2604.02795.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026. URL https://arxiv.org/abs/2604.03128.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3775. URL https://doi.org/10.52202/085713-3775.

Wenjian Zhang, Kongcheng Zhang, Jiaxin Qi, Baisheng Lai, and Jianqiang Huang. Experience is the best teacher: Motivating effective exploration in reinforcement learning for LLMs. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65561.

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. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.18734.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026. URL https://arxiv.org/abs/2605.11182.

A Implementation and Experimental Protocol

A.1 Training Configuration

Table 5 reports the shared training protocol and the ECAR-specific hyperparameters. The initialization checkpoints are `Qwen/Qwen3-1.7B`, `Qwen/Qwen3-4B`, and `Qwen/Qwen3-8B`, with `enable_thinking=False` in the chat template. All three scales use the same scientific configuration; only memory-dependent execution parameters vary with model size. The policy is trained for 200 updates and evaluated every 50 updates.

Table 5: Training and ECAR hyperparameters.

Configuration	Value
Training data	DAPO-Math-17K CleanRef
Offline references	4 GPT-5.4 responses per problem
Online rubricator and verifier	GPT-5.5, high reasoning effort
Query batch size	32
Rollouts per query	8
Training updates	200
Evaluation interval	50 updates
Policy learning rate	1×10^{-6}, constant
Warmup ratio / weight decay	0 / 0.01
PPO mini-batch size	8
PPO micro-batch size per GPU	2
PPO ratio clip, lower / upper	0.2 / 0.28
Dual-clip constant	3.0
Training response length	8192 tokens
Rollout temperature	1.0
Advantage standardization	disabled
Failed-criterion cap K	3
Character similarity / line retention τ	0.70 / 0.70
Edit radius r / context weight ω	8 / 0.25
Baseline factor η / redistribution λ	0.50 / 0.25
Revision temperature / top-p	0.2 / 1.0
Revision response length	8192 tokens

ECAR uses the rubric construction, criterion-wise verification, weighted scoring, and group-relative policy objective of ROPD without modification (Fang et al., 2026). Section 3.1 restates the notation needed to define ECAR. The implementation details below therefore focus on revision certification and advantage redistribution.

Original rollouts truncated at the 8192-token limit retain their rubric scores and standard ROPD updates but are excluded from revision routing. A truncated revision cannot pass the revision certificate.

A.2 Baseline Configurations

Table 6 summarizes the method-specific choices that affect the comparison. The eight RL methods use the same CleanRef training set, batches of 32 queries with eight rollouts each, 1×10^{-6} learning rate, and 200-update budget. DAPO dynamically replenishes informative groups before each update. OPSD and u-OPSD instead follow their released method-native recipes.

For DAPO, the lower/upper clipping parameters are $0.2/0.28$, with dual-clip constant 10. We filter groups with uniform correctness, resampling up to ten batches to collect 32 informative groups. Correct/incorrect outcomes receive $+1/-1$, with a linear overlong penalty over the last 2048 tokens of the 8192-token response budget (maximum penalty 1). GSPO uses lower/upper clipping parameters $0.0003/0.0004$ and averages the token loss within each response, then across responses. Both methods standardize group advantages and disable rollout importance correction; their method-specific settings supplement the shared optimizer and rollout configuration in Table 5.

Table 6: Principal baseline configurations for Qwen3-4B.

Method	Reward	Privileged context	Token credit and key setting
GRPO	Final answer	None	Uniform response advantage
DAPO	Final answer	None	Clip-higher, dynamic sampling, token-mean loss, soft overlong penalty
GSPO	Final answer	None	Sequence-level importance ratio and clipping
ROPD	Weighted rubric	None	Uniform rubric-derived advantage
RLSD	Final answer	Correct final answer	Magnitude reweighting (λ_0=0.5, ε_w=0.2; 50-step decay)
RLCSD	Final answer	Correct and incorrect sibling rollouts	Contrastive reweighting ($K_{\text{neg}} \leq 4$)
CriPO	Weighted rubric	Unexplored or suppressed criteria	UC/SC routing (top three criteria; β=0.03)
ECAR	Weighted rubric	Failed criteria and original response	Certified redistribution (K=3, η=0.5, λ=0.25)

For OPSD, we follow the released Qwen3-4B non-thinking recipe on OpenThoughts Math30K; for u-OPSD, we evaluate the released `u-opsd/qwen3-4b-non-thinking` checkpoint. Both use the benchmark sampling and scoring protocol of Table 1.

A.3 Reference Construction and Model Roles

RawRef contains 17,877 DAPO-Math problems with four initial GPT-5.4 responses per problem. For CleanRef, we first generate four responses at temperature 0.3. If all four are incorrect under the math answer-equivalence evaluator, we generate one additional set of four at temperature 0.7. We retain the first set containing at least one correct answer and discard the problem if neither set qualifies. Retaining a complete generation round gives each of the 16,356 training problems four references.

At every update, the rubricator receives the problem, its four offline references, and the eight current-policy rollouts. It produces a weighted, query-specific criterion list. The verifier evaluates every rollout against the same ordered list and returns both criterion outcomes and a weighted score. For an ECAR candidate, the frozen pre-update policy generates a local revision, which is evaluated against the same rubric. The policy parameters are updated only after all scores, revisions, certificates, and token weights for the batch have been fixed.

A.4 Evaluation Protocol

Evaluation uses temperature 1.0, top-p 0.95, no top-k filtering, and seed 42, with at most 32,768 generated tokens per response. AIME 2024, AIME 2025, HMMT February 2025, and HMMT November 2025 use 16 sampled responses per problem; AMC 2023 uses four. Pass@1 is the mean correctness across these samples. Test Avg. is the unweighted macro average over AIME25, HMMT February/November 2025, and AMC23. For IFEval, we generate 16 responses for each of the 541 prompts. Each response receives one if it satisfies all instructions under the official strict evaluator, and zero otherwise. We average these indicators over all 8,656 responses to obtain prompt-level strict accuracy, reported separately from the mathematical mean. For the shared-budget RL comparison, checkpoints are evaluated at 50-update intervals, and the reported model is selected by AIME24 validation Pass@1 within 200 updates. OPSD and u-OPSD follow their official method-native training configurations; their resulting checkpoints are evaluated with the same benchmark protocol.

Every sampled response remains in the evaluation denominator, including responses stopped at the token limit. Mathematical answers are scored from the generated text using final-answer extraction and equivalence checking; missing or unparseable final answers are incorrect. IFEval applies its strict instruction checks to the generated text.

B PROMPTS AND REVISION CERTIFICATE

B.1 RUBRICATOR AND VERIFIER

We use the ROPD rubricator and verifier templates without changing their instructions or structured output format (Fang et al., 2026). The rubricator constructs criteria from the problem, offline references, and current-policy rollouts. The verifier receives the problem, one response, and the ordered criterion list, and returns one Boolean judgement per criterion together with the weighted rubric score. The same verifier template is used for both the original and revised responses.

B.2 REVISION PROMPT TEMPLATE

The reviser is the frozen pre-update policy π_{θ^-}. Its system prompt is shown below.

ECAR Reviser System Prompt

```
You are a surgical editor of mathematical solutions, not a
fresh solver. Your default action is to copy the current
answer exactly. Silently identify the earliest line or
sentence that must change to satisfy the supplied failed
checks, then apply the smallest causally sufficient edit.

Rules:
1. If the only failure is answer formatting, change only the
   final-answer line.
2. Copy every character before the earliest necessary edit
   verbatim.
3. Replace only the faulty or missing local reasoning.
   Rewrite later text only when it logically depends on that
   repair; otherwise copy it verbatim.
4. Do not improve style, reorganize sections, shorten,
   expand, or replace reasoning that already works.
5. Preserve the original language, headings, notation,
   wording, and formatting outside the necessary repair span.
6. Return the complete revised solution only. Do not
   describe the edit or mention a draft, revision, rubric,
   criterion, check, feedback, evaluator, or extra
   information.

The failed checks are private evidence, not a request to
write a new solution. A mathematically valid but
unnecessarily rewritten answer is a failure.
```

ECAR Reviser User Template

```
[Original problem]
{problem}

[Current answer to edit surgically]
{original response}

[Private failed checks]
{failed criteria}

Copy the current answer and make only the smallest causally
necessary repair. Preserve all unrelated text verbatim.
Output only the complete revised solution.
```

Revision decoding uses temperature 0.2, top-p 1.0, and at most 8192 output tokens. We attempt revision only when the response has between one and three failed criteria and a nonzero rubric advantage.

B.3 Operational Certificate Definitions

The certificate in Equation 2 combines answer correctness, rubric improvement, criterion preservation, generation validity, and locality. Answer correctness is computed by extracting the final answer and applying the same normalized symbolic and numeric equivalence rules used by the math evaluator. Rubric improvement requires $s_i' > s_i$, at least one previously failed criterion to pass, and every previously satisfied criterion to remain satisfied.

For locality, character similarity is the sequence-matching ratio between the complete original and revised strings. Line retention is the fraction of nonempty original lines that occur verbatim in the revision. Both values must be at least the shared threshold τ, set to 0.7 by default. A revision is generation-valid only when it terminates within the configured output budget and does not reproduce the private feedback markers. Failure of any certificate condition leaves the original ROPD update unchanged for that response.

C Algorithm and Alignment Details

C.1 ECAR Training Step

The complete order of operations for one optimizer update is:

1. Freeze the current policy as π_{θ^-} and sample eight responses for each of 32 problems.
2. Construct one weighted rubric per problem and verify all eight responses to obtain s_i, A_i^{R}, and $\mathcal{C}_i^-$.
3. For every eligible response, prompt π_{θ^-} with the problem, original response, and failed criteria to generate y_i'.
4. Verify y_i' against the original rubric and evaluate the revision certificate. Align each certified revision to its original rollout and construct $m_{i,t}$.
5. Teacher-force the original sampled actions under the standard and revision contexts using the same frozen policy. Compute $d_{i,t}$, $u_{i,t}$, and the edit-local token weights $f_{i,t}$.
6. Center and bound the edit-local token weights within each response to obtain $c_{i,t}$, then replace the broadcast advantage with $\widetilde{A}_{i,t}$. Responses without repair support keep $\widetilde{A}_{i,t} = A_i^{\mathrm{R}}$.
7. Apply the standard clipped ROPD/GRPO policy objective and update the policy parameters.

ECAR uses the redistributed token advantages in the standard ROPD policy update, retaining its probability ratios, clipping, token aggregation, and optimizer. The probabilities used to compute the credit weights, and the weights themselves, are treated as constants during each policy update.

C.2 Mapping a Revision to the Original Rollout

ECAR updates the original response, so edits in a certified revision must be mapped to token positions in that response. We deterministically align the original and revised token ID sequences. Original tokens that are replaced or deleted form the edit core E_i. For an insertion, we add the nearest valid original token position to E_i, since the inserted tokens have no direct counterpart in the original response.

We construct the mask $m_{i,t}$ from this edit core using Equation 3. Core positions receive weight 1.0, non-core positions within eight tokens of the core receive weight 0.25, and all other positions receive zero. Overlapping assignments use the maximum weight. Positions with nonzero weight form the repair support S_i. Redistribution requires both a nonempty support and tokens outside it to receive the balancing compensation. If either set is empty, the response retains the original ROPD update.

Illustrative repair. For $2x + 3 = 11$, the target is x^2. The following excerpt illustrates edits within a longer response; unchanged surrounding text is omitted.

Original response	Revised response
Subtract 3: $2x = 8$.	Subtract 3: $2x = 8$.
Divide by 2: $x = \mathbf{3}$.	Divide by 2: $x = \mathbf{4}$.
Next, square x.	Next, square x.
$x^2 = \mathbf{3^2} = \mathbf{9}$.	$x^2 = \mathbf{4^2} = \mathbf{16}$.
The answer is $\mathbf{9}$.	The answer is $\mathbf{16}$.

The arithmetic error requires downstream numerical corrections, while the intervening instruction remains unchanged. If the complete revision passes the certificate, token alignment identifies the actual edits on the original response. Unchanged tokens within eight positions receive neighborhood weight 0.25; more distant unchanged tokens receive zero. Neighborhoods may overlap, and exact boundaries depend on tokenization. Thus support includes repair context, not just erroneous tokens. The revised answer locates edits; policy updates still act on the original rollout rather than treating the revision as an SFT target.

C.3 Ablation Definitions

Joint local JSD. This control uses certified repair evidence for local distillation rather than advantage redistribution. It retains certification and the edit mask, adding a local loss to ROPD with coefficient 1.0. We compute symmetric JSD ($\alpha = 0.5$) between the standard-context student and revision-context teacher distributions over the student's top-100 vocabulary items, grouping the remaining probability mass into one additional category. Following the SRPO implementation used for this baseline, we weight the loss by the edit mask and $\exp(-H_{\text{teacher}})$, where H_{teacher} is the teacher distribution's entropy. This entropy weight reduces the contribution of positions where the teacher is more uncertain and is normalized to mean one over selected positions. Current-to-rollout policy probability ratios are capped at 2.0, and the weighted loss is divided by the number of all valid response tokens in the batch. Gradients flow through the student distribution; teacher probabilities, entropy weights, and probability ratios are treated as constants during each update.

Dense ungated. This variant removes revision certification and the edit mask, evaluating revision drop over the full original response. Instead of separating repair support from the tokens used for compensation, it centers the drop over all valid token positions V_i:

$$\hat{c}_{i,t}^{\text{dense}} = (1-\eta)\left(u_{i,t} - \frac{1}{|V_i|}\sum_{k\in V_i} u_{i,k}\right), \qquad t \in V_i. \tag{10}$$

This replaces Equation 7 with the centered form of $\eta + (1-\eta)u_{i,t}$. The subsequent bounding and advantage update remain the same as in ECAR.

Random support. This control retains certification and the number and values of nonzero edit-mask weights, but changes their locations. For each certified response, we move the weights to a contiguous span of the same size as the original support, sampling its start uniformly over valid positions. Core weights are placed in the center, with neighborhood weights split between the two sides. We then evaluate revision drop at the new positions and compute local weights using Equation 6.

Support-averaged drop. This variant retains certification, the edit mask $m_{i,t}$, and the redistribution rule, while removing tokenwise variation in revision drop within the support. Each drop is replaced by the average $\bar{u}_i$ weighted by the edit mask:

$$\bar{u}_i = \frac{\sum_{k\in S_i} m_{i,k}u_{i,k}}{\sum_{k\in S_i} m_{i,k}}, \qquad f_{i,t}^{\text{control}} = m_{i,t}[\eta + (1-\eta)\bar{u}_i] = m_{i,t}\frac{\sum_{k\in S_i} f_{i,k}}{\sum_{k\in S_i} m_{i,k}}. \tag{11}$$

The drop is shared across the support, while core and neighborhood weights remain distinct. The sum of local weights, total absolute redistribution mass per response, and λ match ECAR.

D Computational Structure

Relative to ROPD, ECAR adds computation only for revision-eligible responses. It generates one local revision, verifies that revision against the existing rubric, and evaluates the original sampled actions under the failed-criterion context. Responses that are ineligible or fail certification use the original ROPD path, and no ECAR-specific loss is evaluated for them. All main experiments use eight H200 GPUs; the rubricator and verifier are queried through the same online service throughout all compared runs.

E Additional Optimization and Mechanism Analysis

E.1 Signal Normalization and Support Concentration

Figures 1(a) and 3 compare all signals on the same 297 responses from 144 mathematical problems. We construct this analysis set by sampling eight responses per problem from 512 problems, yielding 4,096 responses. After excluding one problem with an incomplete verifier output, we obtain 396 certified revisions across 196 problems.

We then retain responses with nonempty repair support that meet the reference requirements of both OPSD and RLCSD. OPSD requires a correct GPT-5.4 reference. RLCSD requires at least one correct and one incorrect response among the other responses to the same problem.

Using the same frozen Qwen3-4B before method-specific training, we evaluate each original token prefix under the contexts required by each method.

OPSD compares the standard context with a context containing a correct GPT-5.4 reference, selected uniformly from the compatible references. Its token score is the forward KL between the two distributions. RLCSD instead uses contexts containing correct or incorrect responses to the same problem (Pan et al., 2026). We compare the sampled token's probability under one correct response with its mean probability under up to four incorrect responses. The score is the absolute difference between the logarithms of these two probabilities.

The Revision drop row uses the bounded score $u_{i,t}$ from Equation 5. This score measures how much the revision context lowers an original sampled token's probability relative to the standard context. We evaluate it over the full response without applying the repair-support mask.

The ECAR row shows the edit-local weights $f_{i,t}$ from Equation 6. These weights combine the repair-support mask with the bounded revision-drop score and are zero outside the support.

Figure 1(a) measures how strongly the OPSD signal is concentrated around a successful repair. For each response, we divide the mean forward KL within its repair support by the mean over the full response, then report the median of these ratios.

Figure 3(a) compares the share of signal mass inside the repair support. For each response, we divide the sum of scores inside the support by the sum over all valid tokens, then average across responses. Let $a_{i,t}^{(m)} \geq 0$ be method m's score for token t of response i, with repair support S_i and valid positions V_i. For N responses,

$$r_i^{(m)} = \frac{\sum_{t \in S_i} a_{i,t}^{(m)}}{\sum_{t \in V_i} a_{i,t}^{(m)}}, \qquad \bar{r}^{(m)} = \frac{1}{N} \sum_{i=1}^{N} r_i^{(m)}. \tag{12}$$

The outside-support fraction is $1 - r_i^{(m)}$ and is averaged in the same way. Uniformly distributing mass over valid tokens gives an inside-support share of $N^{-1} \sum_i |S_i|/|V_i| = 7.52\%$.

The ECAR row shows the masked weights $f_{i,t}$, with 100% of their mass inside repair support and zero outside. The other three signals are unmasked, so their shares measure alignment with successful repairs. ECAR's policy update also includes a uniform adjustment outside the support to preserve the mean advantage. The final coefficient $c_{i,t}$ thus has equal absolute mass inside and outside for each nonzero redistribution: $\sum_{t \in S_i} |c_{i,t}| = \sum_{t \in U_i} |c_{i,t}|$.

E.2 Token Classes

Figure 3(b) divides tokens into three classes using RLCSD's lexical rules (Pan et al., 2026). After normalizing tokenizer boundary markers and lowercasing each decoded token, we first check for *task* content: a digit, mathematical operator, inequality, LaTeX command, or a mathematical term such as `sqrt`, `frac`, `prime`, or `gcd`. If none is present, we assign *style* to tokens consisting of punctuation, a line break, a discourse marker such as `therefore`, `however`, or `finally`, or a function word. Tokens matching neither class are labeled *other*. The same token labels are used for all four signals on the common analysis set.

For each response, we sum the scores in each class and divide by the total score over valid tokens. We then average these shares across responses. Writing $z_{i,t}$ for the token class, the share for $c \in \{\text{task}, \text{style}, \text{other}\}$ is

$$q_{i,c}^{(m)} = \frac{\sum_{t \in V_i : z_{i,t}=c} a_{i,t}^{(m)}}{\sum_{t \in V_i} a_{i,t}^{(m)}}, \qquad \bar{q}_c^{(m)} = \frac{1}{N} \sum_{i=1}^{N} q_{i,c}^{(m)}. \tag{13}$$

The reported style-to-task ratio is $\bar{q}_{\text{style}}^{(m)} / \bar{q}_{\text{task}}^{(m)}$.

E.3 Optimization Diagnostics

Figure 4 compares GRPO, DAPO, GSPO, ROPD, RLSD, RLCSD, CriPO, Joint local JSD, and ECAR. Joint local JSD is the ablation in Table 3 that augments ROPD with a local JSD loss on certified repair support. Panel (a) tracks AIME24 validation Pass@1. Panels (b) and (c) separate answer-correctness rewards from weighted rubric scores; panels (d–f) show response length, truncation rate, and gradient norm. Training traces use ten-update moving averages. DAPO's traces describe the update batches retained by dynamic sampling.

ECAR leads in late-stage validation performance while its response lengths and gradient norms remain close to those of ROPD and Joint local JSD. The outcome-based methods improve their training rewards but reach lower validation scores. CriPO's post-50 decline in validation performance coincides with a decrease in rubric reward and larger fluctuations in response length and truncation rate.

F Additional Main-Result Checks

F.1 Question-Paired Confidence Intervals

We estimate uncertainty in ECAR's Qwen3-4B test gains with a paired bootstrap. Each replicate resamples questions with replacement within each of the four mathematical test benchmarks, using the same questions for ECAR and the baseline. We then average their score differences across these four benchmarks. From 10,000 replicates, the 95% confidence interval for ECAR's gain over ROPD is $[0.23, 4.95]$ percentage points. Its gain over CriPO is 15.83 points, with a 95% confidence interval of $[12.43, 19.41]$. Its gains over DAPO and GSPO are 11.51 and 12.59 points, with intervals of $[8.55, 14.69]$ and $[8.98, 16.51]$, respectively. The intervals for its gains over GRPO, RLCSD, and RLSD also lie above zero (Figure 5).

F.2 Common Step-200 Comparison

Table 7 compares Qwen3-4B methods after 200 training steps. ECAR leads on all four mathematical test benchmarks, reaching a Test Avg. of 59.59 versus 55.34 for ROPD. The main-result gains therefore also hold at a common training endpoint.

F.3 Results Across Independent Runs

Table 8 compares three independent Qwen3-4B runs each of ECAR and CleanRef ROPD at step 200, including those in Table 7. Averaged across runs, ECAR scores higher on every mathematical test benchmark. Its Test Avg. exceeds ROPD's in all three comparisons, with an average gain of 3.39

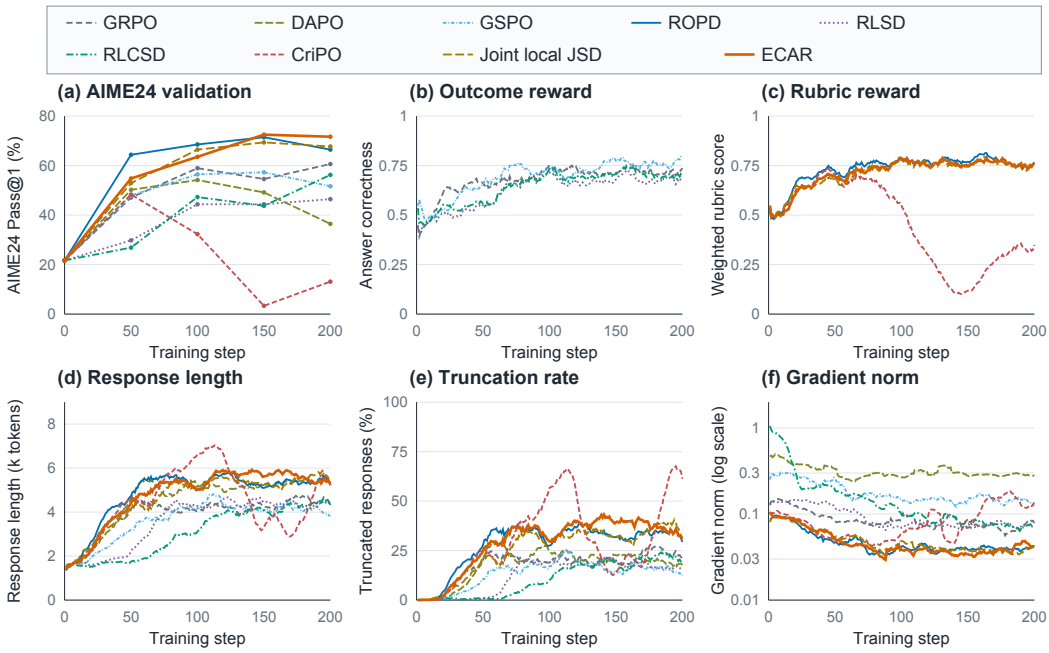

Figure 4: **Validation performance and optimization diagnostics.** (a) AIME24 validation Pass@1. (b) Answer-correctness reward for GRPO, GSPO, RLSD, and RLCSD. (c) Weighted rubric score for ROPD, CriPO, Joint local JSD, and ECAR. (d) Response length. (e) Truncation rate. (f) Gradient norm on a logarithmic axis. Panels (b–f) use ten-update moving averages. Joint local JSD denotes ROPD with auxiliary local distillation on certified repair support.

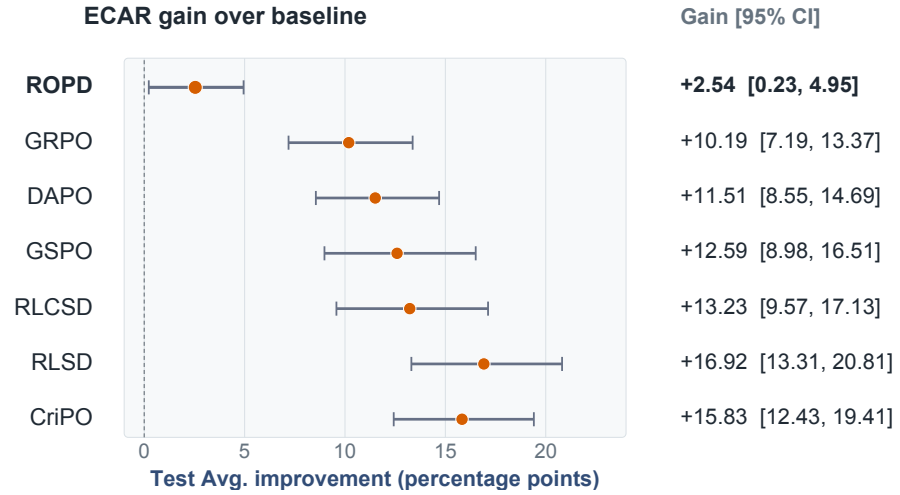

Figure 5: **Question-paired confidence intervals for ECAR's gains.** Points show ECAR's improvement in Test Avg. over each baseline. Bars show percentile 95% confidence intervals from 10,000 question-paired bootstrap replicates.

percentage points. The sample standard deviation of Test Avg. is 0.39 for ECAR versus 1.45 for ROPD.

Table 7: **Qwen3-4B results at the common step-200 endpoint.** Initial is the step-0 model; all trained methods use step 200. ROPD uses CleanRef. Scores are percentages. Test Avg. averages the four math test benchmarks; IFEval is reported separately.

Method	Step	AIME24 (val)	AIME25	HMMT-F	HMMT-N	AMC23	Test Avg.	IFEval
Initial	0	21.67	18.33	10.42	6.46	61.45	24.16	80.14
GRPO	200	60.63	48.54	27.50	39.17	83.13	49.59	80.42
DAPO	200	36.46	33.96	21.88	29.17	83.43	42.11	80.34
GSPO	200	51.67	52.29	22.71	32.50	80.72	47.06	81.39
ROPD	200	66.46	56.04	34.38	44.79	86.14	55.34	81.13
RLSD	200	46.46	41.67	22.71	28.13	78.92	42.85	80.09
RLCSD	200	56.25	48.54	23.75	34.38	79.52	46.55	80.70
CriPO	200	13.13	10.42	6.25	2.29	37.65	14.15	**81.85**
ECAR	200	**71.67**	**62.71**	**39.79**	**47.92**	**87.95**	**59.59**	81.02

Table 8: **Qwen3-4B results across three independent runs at step 200.** Scores are percentages, reported as means $\pm$ sample standard deviations across runs. Test Avg. averages the four math test benchmarks.

Method	AIME24 (val)	AIME25	HMMT-F	HMMT-N	AMC23	Test Avg.
ROPD	68.13 ± 1.50	57.43 ± 2.06	36.32 ± 2.23	46.18 ± 1.36	86.65 ± 0.46	56.64 ± 1.45
ECAR	$\mathbf{72.36 \pm 1.58}$	$\mathbf{63.06 \pm 0.43}$	$\mathbf{41.88 \pm 1.91}$	$\mathbf{47.85 \pm 0.12}$	$\mathbf{87.35 \pm 0.80}$	$\mathbf{60.03 \pm 0.39}$