TRUER: TEST-TIME REWARD CORRECTION USING AN ERROR-AWARE RECTIFIER FOR REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time reinforcement learning enables language models to adapt using rewards derived from their own answers. However, majority voting can reinforce incorrect solutions even when correct alternatives are available. Our training audit shows that false positives—incorrect responses receiving positive rewards—dominate reward errors. We focus on a practical regime where ground-truth rewards are difficult to obtain at scale, while a limited labeled source set is available. We introduce TRUER, a framework that uses limited supervision to train a reward rectifier that distinguishes correctness from consensus. To achieve this goal, we train the rectifier with an emphasis on incorrect majorities and correct minorities, targeting cases where consensus and correctness disagree. After that, we freeze the rectifier and use it to correct rewards during subsequent test-time adaptation. By improving reward accuracy and thereby model performance, TRUER consistently outperforms naive TTRL in both metrics across all evaluated configurations.

1 INTRODUCTION

Reliable rewards are central to reinforcement learning for language model reasoning (Shao et al., 2024; DeepSeek-AI et al., 2025). Yet ground-truth rewards can be difficult to obtain at scale. Test-time reinforcement learning (TTRL) addresses this challenge by deriving pseudo-rewards from the model's own responses (Zuo et al., 2025). A common approach samples several solutions and uses the most frequent answer as a training target, extending the idea of self-consistency from inference to learning (Wang et al., 2022).

But consensus does not ensure correctness. Even when a correct answer appears among the model's 32 sampled responses, majority voting can favor an incorrect answer, as illustrated by the gap between Pass@32 and Majority@32 in Figure 2(a). Majority-based rewards can then reinforce an incorrect majority while withholding reward from a correct minority. The challenge is therefore to distinguish correct answers from false consensus and use them to guide subsequent learning.

Ground-truth answers provide direct supervision for this distinction, but obtaining them at scale for target problems can be costly. Prior work uses limited supervision to assess pseudo-label quality (Li et al., 2024) and guide reinforcement learning on unlabeled problems (Yang et al., 2025b). We therefore focus on a practical regime with limited labeled source data, disjoint from the target adaptation questions, while target ground-truth rewards remain unavailable. We investigate how this source supervision can help correct erroneous consensus without accessing target ground-truth answers.

To this end, we introduce **T**est-Time **R**eward Correction **U**sing an **E**rror-Aware **R**ectifier (**TRUER**), which turns limited source supervision into reusable reward correction. The rectifier is trained on candidate solutions organized by both correctness and consensus membership, with emphasis on incorrect majorities and correct minorities. After training, it is frozen. During test-time adaptation, the rectifier verifies the leading answer, checks a second-ranked alternative after explicit rejection, and abstains when neither provides a reliable target. This design aims to improve reward accuracy while retaining useful learning signals.

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.

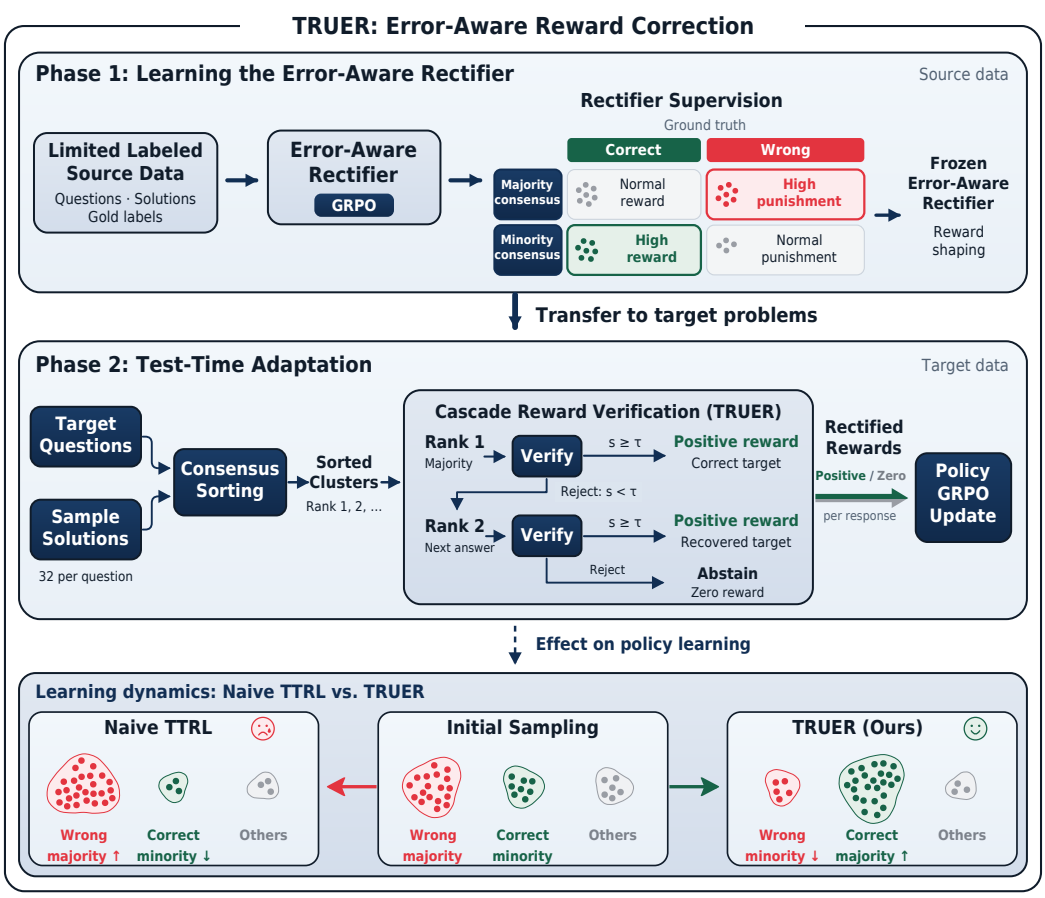

Figure 1: **TRUER overview and learning dynamics.** Top: limited source labels train an error-aware rectifier, which is then frozen to correct rewards during test-time adaptation. Bottom: starting from the same sampled answer distribution, naive TTRL can reinforce a wrong majority, whereas TRUER can recover a correct minority and support policy learning.

Our experiments evaluate reward quality and final model performance across several mathematical reasoning configurations, comparing TRUER with naive TTRL and an oracle reference trained with ground-truth rewards using GRPO. Our contributions are threefold.

- **Reward reliability in majority-vote TTRL.** We analyze the reliability of majority-based rewards in TTRL. Despite high Pass@32, consensus can miss correct candidates; our training audit further shows that reward accuracy does not improve consistently and that false positives dominate reward errors.

- **Limited-supervision reward rectification.** We introduce **TRUER**, which emphasizes incorrect majorities and correct minorities when training a rectifier on a disjoint labeled source set. The frozen rectifier rejects incorrect consensus, recovers alternatives, and abstains when needed, correcting rewards without querying target ground truth.

- **Empirical evaluation and analysis.** Across all evaluated configurations, TRUER improves reward accuracy and final model performance over naive TTRL, narrowing the gap to GRPO trained with ground-truth rewards. Reward audits, training trajectories, and ablations further examine how reward correction changes supervision and supports policy learning.

2 Related Work

Test-time learning with self-generated rewards. TTRL (Zuo et al., 2025) uses majority voting to estimate labels for reinforcement learning. Its "lucky hit" effect can maintain high reward accuracy because incorrect answers differing from an incorrect pseudo-label still receive correct zero rewards. However, consensus does not ensure correctness, and rewarding incorrect consensus may reinforce the model's existing reasoning errors.

Pseudo-label selection through consistency and reflection. Self-Harmony (Wang et al., 2025) uses a reframer to reformulate questions, then selects answers receiving consistent support across original and reframed versions. SR-TTRL (Wu et al., 2026) instead groups and summarizes candidate solutions, then prompts the model to compare their reasoning and select a pseudo-label. Meanwhile, SCRL (Yan et al., 2026) restricts positive rewards to sufficiently strong consensus and uses entropy-gated negative feedback to suppress unlikely answers. These methods derive corrective signals from the model's answer distributions or self-assessment. When these signals share the model's reasoning errors, they may provide insufficient evidence to reject incorrect majorities or recover correct minorities.

Verification training with consensus-derived supervision. CoVerRL (Pan et al., 2026) jointly trains generation and verification: consensus-derived contrastive examples supervise the verifier, whose judgments then filter pseudo-labels. Similarly, TTRL-CoCoV (Li et al., 2026) uses high-confidence majority answers to train verification and applies the resulting judgments to select pseudo-labels for low-confidence problems. Although this interaction can improve supervision, verification targets still depend on model consensus. Consequently, incorrect consensus may propagate into verification training, reinforcing existing biases and limiting corrective feedback.

Reward correction with limited supervision. CARE (Wang et al., 2026) selects problems for annotation using estimated reliability and correction value, but requires ground-truth access during adaptation. Despite labeled supervision, its acquisition classifiers use policy-derived features that may reflect model biases. Our method instead trains a reward rectifier on a labeled source set disjoint from the target problems, emphasizing incorrect majorities and correct minorities. The rectifier is then frozen and reused to correct rewards without acquiring ground-truth labels during target adaptation.

3 When Consensus Fails to Provide Reliable Supervision

3.1 Correct Solutions Can Be Missed by Consensus

Candidate coverage and answer selection are distinct aspects of inference-time scaling (Brown et al., 2024). For each question, Pass@32 measures whether a correct answer appears in a fixed pool of sampled responses, whereas Majority@32 measures whether voting selects it. Their difference identifies questions on which a correct candidate is available but overlooked. Figure 2(a) shows this gap across the inspected naive-TTRL endpoints. It is particularly pronounced for the smaller Qwen3 model on AMC23. Correct solutions are reachable, but answer frequency does not reliably identify them.

This gap motivates checking correctness before turning a popular answer into supervision, consistent with the logical-minority diagnosis of Wu et al. (2026).

3.2 False Positives Dominate Residual Reward Errors

Let a_i be the answer extracted from response y_i, m the most frequent answer, and a^* the ground-truth answer. We use majority to mean the largest answer cluster, which need not contain more than half the responses. Majority and correctness rewards are

$$r_i^{\mathrm{mv}} = \mathbb{1}[a_i \equiv m], \qquad r_i^{*} = \mathbb{1}[a_i \equiv a^*], \tag{1}$$

where $\equiv$ denotes mathematical equivalence. A false positive (FP) rewards an incorrect response; a false negative (FN) withholds reward from a correct response. Overall reward accuracy measures agreement between these binary rewards and true correctness.

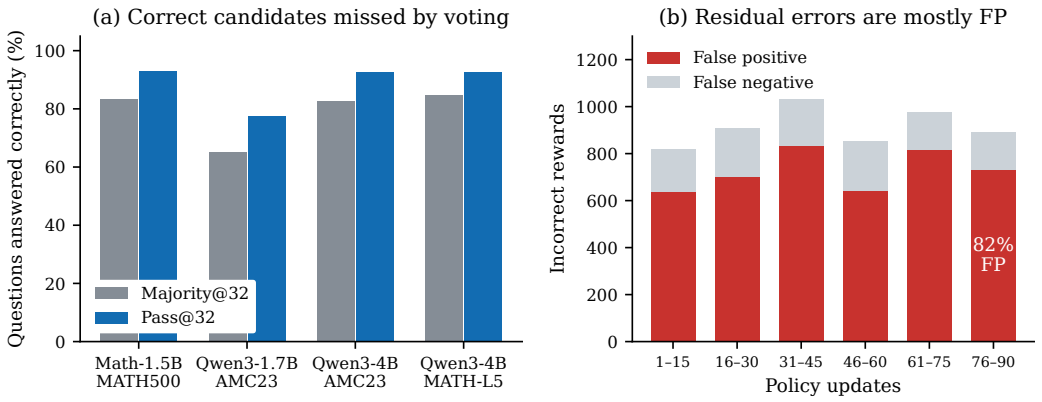

Figure 2: **Available correct answers do not guarantee reliable rewards.** (a) Pass@32 and Majority@32 use the same candidate pool at each naive-TTRL endpoint. The gap represents correct candidates missed by voting. (b) In the Qwen2.5-Math-1.5B/MATH500 training audit, false positives dominate reward errors across disjoint training windows, including late training.

Our MATH500 audit shows that continued training does not consistently improve reward accuracy. Across consecutive 15-update windows, accuracy fluctuates and ends at 94.21%, below the initial 94.67%. Meanwhile, false positives dominate reward errors in every window, accounting for 82% in the final window (Figure 2(b)). Incorrect answers outside the selected cluster still receive the correct zero reward, keeping aggregate accuracy high despite these erroneous positive rewards.

3.3 Implications for Reward Correction

With group-relative policy optimization (GRPO), a rewarded response receives positive outcome advantage when its group contains mixed rewards (Shao et al., 2024). A wrong majority can therefore be reinforced while a correct minority is discouraged. Constant-reward groups have no such outcome contrast.

For example, suppose that 18 of 32 responses share an incorrect answer, 10 give the correct answer, and the remaining four give other incorrect answers. Majority voting assigns reward 1 to the 18 incorrect responses and reward 0 to the correct minority. Under GRPO, this reward contrast can increase the probability of the incorrect answer, making it more likely to dominate subsequent samples. Repeated updates can thus reinforce the wrong majority and make the correct answer harder to recover.

A useful correction policy must address both sides of this failure. Rejecting an incorrect majority removes false-positive rewards; accepting a correct alternative restores a learning target. Rejection alone may leave no outcome signal. The bottom panel of Figure 1 illustrates these contrasting learning dynamics. We therefore evaluate reward accuracy together with correct-reward retention, active training groups, and model performance.

4 Method

4.1 Problem Setup and Overview

We assume a labeled source set $\mathcal{D}_s = \{(x, a^*)\}$ and a target set $\mathcal{D}_t$ whose ground-truth answers are unavailable for reward computation. Rectifier fitting and subsequent test-time adaptation use disjoint sets of questions. TRUER first trains a rectifier V_ϕ on source candidates, then freezes ϕ while adapting a language model policy π_θ on target inputs (Figure 1).

Table 1: **TRUER target adaptation.** The rectifier is trained on source labels before this procedure.

Input: policy π_θ, frozen rectifier V_ϕ, target questions $\mathcal{D}_t$.
For each policy update:

1. Sample a batch of target questions and candidate solutions from π_θ.
2. Group equivalent answers and rank the clusters by frequency.
3. Initialize $\hat{a} = \perp$. Verify rank1 when its route is eligible and untied.
4. If rank1 is accepted, select it. If explicitly rejected, verify an eligible, untied rank2 and select it only if accepted.
5. Assign rewards using Equation 3 and update θ with GRPO.

Output: adapted policy and the unchanged rectifier.

4.2 Learning the Reward Rectifier

Each rectifier example contains a question, a candidate solution, and its extracted answer. Source ground truth determines correctness, while the sampled candidate pool determines majority membership. Their cross-product gives four categories: majority-correct, majority-wrong, minority-correct, and minority-wrong. We emphasize the two categories where consensus and correctness disagree, while retaining examples of correct majorities and incorrect minorities.

The rectifier produces a short justification and an integer score $s \in \{0, \ldots, 9\}$. With $p = s/9$ and correctness label $c \in \{0, 1\}$, its training reward is

$$R_V = 1 - 2(p - c)^2 - 0.1\,\mathbb{1}[\text{explicit reason–score conflict}]. \tag{2}$$

Malformed or truncated outputs receive a fixed penalty. GRPO optimizes the rectifier using multiple judgments per candidate. Category weights emphasize incorrect majorities and correct minorities; they are applied to the policy loss after within-group advantage normalization. Neither the gold answer nor the category label is included in the rectifier input. The weighting formula and recorded training settings are given in Appendix A.3.

4.3 Rectified Rewards for Target Adaptation

The model samples responses and groups equivalent final answers. TRUER checks an eligible representative of the largest cluster, rank1. A valid score $s \geq \tau$ accepts that answer, with threshold $\tau = 5$ in our experiments. An explicit rejection ($s < \tau$) triggers a rank2 check; an unresolved judgment does not. If neither branch selects an answer, TRUER abstains. Ties and candidate eligibility are handled by fixed rules described in Appendix A.3.

For selected answer $\hat{a}$, with $\perp$ denoting abstention, the reward and outcome advantage are

$$r_i^{\mathrm{TRUER}} = \mathbb{1}[\hat{a} \neq \perp]\mathbb{1}[a_i \equiv \hat{a}], \qquad A_i = \frac{r_i^{\mathrm{TRUER}} - \bar{r}^{\,\mathrm{TRUER}}}{\mathrm{std}(r^{\mathrm{TRUER}}) + 10^{-6}}. \tag{3}$$

The policy is updated with GRPO while the rectifier remains frozen (Table 1). Abstention assigns all-zero rewards and removes the group's direct outcome advantage.

5 Experiments

5.1 Experimental Setup

Models and tasks. We evaluate Qwen2.5-Math-1.5B and Qwen3 models with 1.7B or 4B parameters (Yang et al., 2024; 2025a), as well as Llama-3.1-8B-Instruct (Grattafiori et al., 2024), using Qwen3 rectifiers with 1.7B or 4B parameters. Table 2 contains nine configurations across MATH500 and MATH-L5, both drawn from MATH (Hendrycks et al., 2021), as well as AMC23 and AIME24. The MATH500 experiments reserve 130 questions for rectifier development and use the remaining 370 for target adaptation and evaluation. MATH-L5 uses a separate set of 600 questions. For Qwen models on AMC23 and AIME24, rectifiers use source questions from AMC2024

Table 2: **TRUER main results (%).** Columns report the model/task, rectifier, Pass@1, Pass@32, and reward accuracy. In each Pass@1 bar, the gray line spans the TTRL–Oracle gap and the blue line shows the TTRL–TRUER gain. Labels give the gain in percentage points (above), and the full gap in percentage points and the fraction closed (below).

Model / task	Rectifier	Pass@1				Pass@32			Reward acc.	
		Base	TTRL	TRUER	Oracle	TTRL	TRUER	Oracle	TTRL	TRUER
Qwen2.5-Math 1.5B / MATH500	Qwen3-4B	21.85	69.62	**70.46**	70.87	92.97	**91.89**	93.24	95.53	**96.36**
				+0.84 pp						
			TTRL	1.25 pp / 67.6%	Oracle					
Qwen3-1.7B AMC23	Qwen3-1.7B	46.09	56.09	**57.11**	64.61	77.50	**92.50**	92.50	74.84	**86.09**
				+1.02 pp						
			TTRL	8.52 pp / 11.9%	Oracle					
Qwen3-1.7B AMC23	Qwen3-4B	46.09	56.09	**59.38**	64.61	77.50	**92.50**	92.50	74.84	**89.22**
				+3.28 pp						
			TTRL	8.52 pp / 38.5%	Oracle					
Qwen3-1.7B AIME24	Qwen3-1.7B	13.96	20.21	**21.46**	24.69	46.67	**50.00**	46.67	61.41	**86.80**
				+1.25 pp						
			TTRL	4.48 pp / 27.9%	Oracle					
Qwen3-1.7B AIME24	Qwen3-4B	13.96	20.21	**24.79**	24.69	46.67	**43.33**	46.67	61.41	**88.28**
				+4.58 pp						
			TTRL	4.48 pp / 102.3%	Oracle					
Qwen3-4B AMC23	Qwen3-4B	65.39	79.69	**82.34**	85.63	92.50	**97.50**	97.50	87.73	**97.66**
				+2.66 pp						
			TTRL	5.94 pp / 44.7%	Oracle					
Qwen3-4B MATH-L5	Qwen3-4B	69.43	79.82	**80.68**	83.76	92.50	**93.00**	94.83	87.81	**89.92**
				+0.86 pp						
			TTRL	3.94 pp / 21.8%	Oracle					
Llama-3.1-8B Instruct / MATH500	Qwen3-4B	37.02	63.93	**73.61**	83.68	67.03	**85.95**	91.62	75.66	**96.12**
				+9.69 pp						
			TTRL	19.76 pp / 49.0%	Oracle					
Llama-3.1-8B Instruct / AMC23	Qwen3-4B†	12.42	49.45	**60.23**	88.20	50.00	**75.00**	95.00	61.48	**89.48**
				+10.78 pp						
			TTRL	38.75 pp / 27.8%	Oracle					

†Same rectifier as the Llama/MATH500 configuration above.

and AIME2025, respectively. Llama on AMC23 reuses a frozen rectifier trained solely on MATH. Detailed experimental protocols are provided in Appendix A.

Comparisons and metrics. **Naive TTRL** implements the majority-vote reward mechanism of Zuo et al. (2025) under our experimental protocol; **TRUER** uses rectified rewards; **Oracle** is a supervised reference using target ground-truth rewards with GRPO (Shao et al., 2024) under the corresponding policy training configuration. Base is the model before adaptation. Pass@1 estimates single-sample accuracy from the saved responses, Pass@32 measures candidate coverage, and Majority@32 measures voting accuracy on the same pool (Chen et al., 2021). Reward accuracy is the fraction of binary rewards agreeing with response correctness, aggregated over the training windows specified in Appendix A. The main table reports each model's final checkpoint; the mechanism audit instead compares reward policies on one fixed response pool.

Evaluation scope. All reported scores are measured in our own experiments, including the Base, naive TTRL, and Oracle references. We evaluate model performance on the target adaptation sets. Target labels are excluded from TRUER's reward decisions.

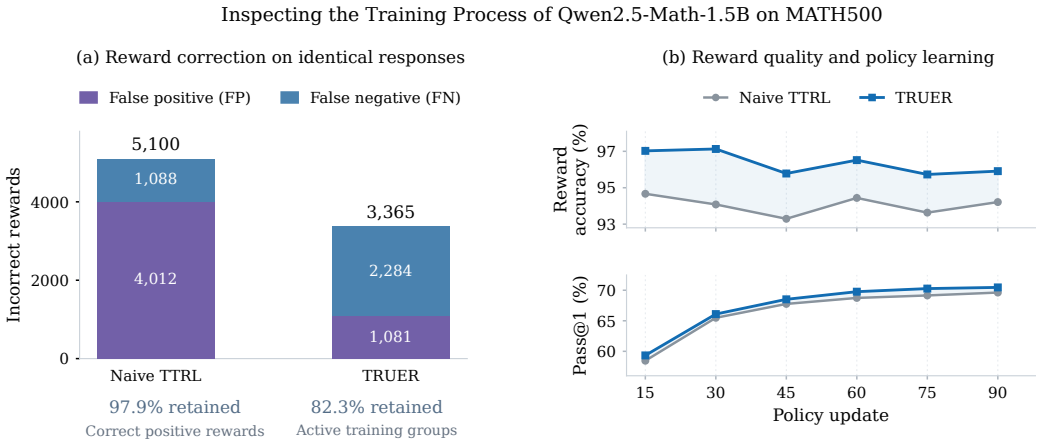

Figure 3: **Reward correction on MATH500.** Qwen2.5-Math-1.5B with a Qwen3-4B rectifier. (a) FP/FN counts on identical responses; retention is relative to majority voting. (b) Reward accuracy over 15-update windows and endpoint Pass@1 on each method's own trajectory.

5.2 Main Results

Across all nine configurations in Table 2, TRUER improves reward accuracy and final Pass@1 over naive TTRL, narrowing the Pass@1 gap to the Oracle reference trained with ground-truth rewards. For example, using a Qwen3-4B rectifier, Qwen2.5-Math-1.5B on MATH500 improves reward accuracy from 95.53% to 96.36% and closes 67.6% of the TTRL–Oracle gap; Qwen3-1.7B on AMC23 improves reward accuracy from 74.84% to 89.22% and closes 38.5% of the gap. The final Llama-3.1-8B-Instruct/AMC23 row reuses a MATH-trained rectifier without AMC-specific rectifier training, suggesting some transferability across datasets.

Pass@32 varies less consistently. Where it changes little despite higher Pass@1, reinforcement learning may sharpen the answer distribution toward correct solutions already reachable by the base model, improving sampling efficiency without necessarily expanding its reasoning coverage. This interpretation is consistent with prior observations on RLVR (Yue et al., 2025).

5.3 Case Study: Reward Correction and Model Adaptation

We investigate Qwen2.5-Math-1.5B with a Qwen3-4B rectifier on MATH500 to examine how reward correction relates to model adaptation.

Reward accuracy and model performance. We first apply naive majority rewards and TRUER to the same 92,160 responses from the TRUER training run. As Figure 3(a) shows, total reward errors fall from 5,100 to 3,365, with 73.1% fewer false positives, although false negatives increase. TRUER retains 97.9% of the correct-positive reward count, including correct rewards recovered through rank2, and 82.3% of active training groups relative to majority voting. Vote-margin filters retain fewer correct rewards at comparable false-positive removal (Appendices C and C.2).

Across training, TRUER achieves higher reward accuracy in every 15-update window and higher Pass@1 at all six checkpoints (Figure 3(b)). Appendix B provides additional learning curves and uncertainty estimates.

A concrete case: correct minority becomes majority. Figure 4 tracks both methods on the same question at matched training steps. TRUER rejects an incorrect majority and rewards the correct minority, which later becomes dominant, while naive TTRL retains the wrong consensus. Recovery is not guaranteed: repeated abstention can leave the model incorrect. Appendix D gives both examples and their selection rules, while Appendix B.3 reports aggregate answer-distribution changes.

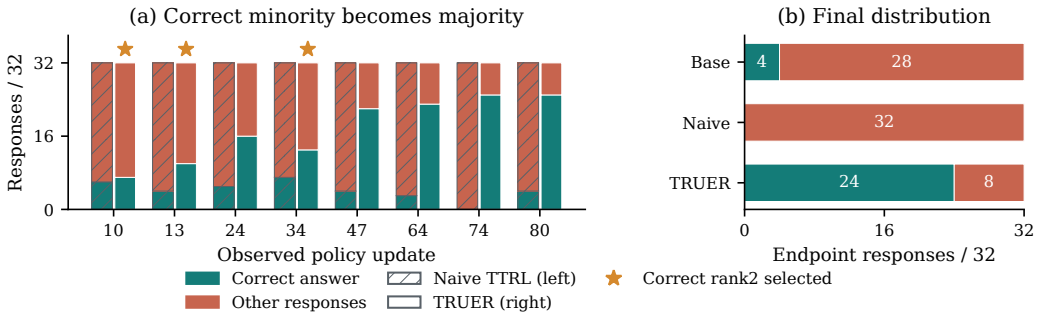

Figure 4: **From correct minority to majority.** (a) Each training visit pairs naive TTRL (left, hatched) with TRUER (right, solid); stars mark correct rank2 selection. (b) Separately sampled endpoint responses. Each bar contains 32 responses; "Other responses" includes incorrect and unparsed answers.

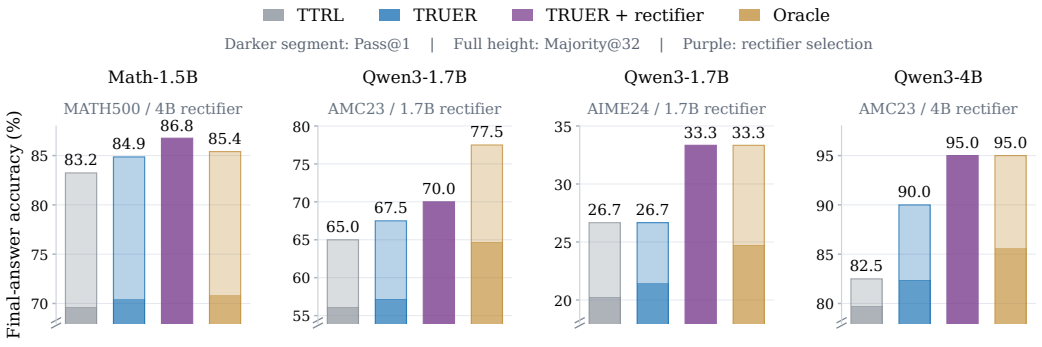

Figure 5: **Rectifier reuse at deployment.** For TTRL, TRUER, and Oracle, darker segments end at Pass@1 and lighter extensions reach Majority@32. Purple bars show TRUER with score-weighted answer aggregation on identical 32-candidate pools. Panels use different truncated y-axis ranges. Four configurations are shown; Appendix E reports all six.

5.4 Additional Evaluation: Rectifier Reuse at Deployment

Scoring candidate solutions is a common use of test-time computation (Cobbe et al., 2021; Snell et al., 2024). We therefore evaluate whether the rectifier trained for reward correction can also support answer selection at deployment. Given 32 sampled candidates, we group equivalent answers and sum their thresholded rectifier scores, normalized by the square root of the number of valid judgments in each group. We select the answer with the largest aggregate support, with majority fallback if no candidate passes. Figure 5 compares this strategy with single-sample answering and majority voting for naive TTRL, TRUER, and Oracle. Under this shared rule, selected through exploratory target-set replay, rectifier selection improves over TRUER majority voting in all six evaluated configurations, by 0.17–6.67 percentage points. With 32 candidates, it exceeds Oracle single-sample accuracy in all six configurations. Appendix E specifies the aggregation rule, its selection procedure, matched naive-TTRL results, and inference cost.

5.5 Ablations and Limitations

Effect of rectifier training. In our fixed-pool comparisons, the trained rectifier achieves higher judgment accuracy and rejects more incorrect answers than its pretrained counterpart. For example, on responses generated by naive TTRL, training increases judgment accuracy from 71.85% to 86.52% and reduces wrong-answer acceptance from 52.11% to 18.35%. Detailed results and evaluation settings are provided in Appendix C.4. For adaptation, TRUER with a trained rectifier achieves higher final Pass@1 than its untrained-rectifier variant in the three comparisons reported in Table 14.

On AIME24 with the 1.7B rectifier, the untrained variant even falls below naive TTRL (18.96% vs. 20.21%), whereas the trained variant reaches 21.46%.

Training overhead. Although deploying the adapted model alone incurs no additional rectifier computation, reward correction increases training time. For Qwen2.5-Math-1.5B on MATH500, the recorded mean time per policy update increases from 177.2 seconds for naive TTRL to 219.3 seconds for TRUER, an overhead of approximately 23.7% with a dedicated rectifier GPU, excluding rectifier pretraining (Appendix F).

Training-budget controls. We extend naive TTRL to approximately match TRUER's adaptation-stage GPU-hours, and separately extend TRUER to match the original naive TTRL run's cumulative number of groups with nonzero outcome advantages, to assess TRUER's performance relative to naive TTRL under comparable computation and cumulative counts of advantage-bearing groups, respectively. Additional naive training narrows the gap in some settings, but TRUER retains higher final Pass@1 in all five GPU-hour-matched comparisons. The active-group-matched controls likewise preserve TRUER's advantage over the original naive TTRL checkpoints. Full results and protocols are provided in Appendix F.

Evaluation scope. Our evaluation focuses on mathematical reasoning; extending reward rectification to open-ended tasks with less structured answers remains an interesting direction.

6 Conclusion

We introduced TRUER, which uses limited source supervision to correct unreliable consensus rewards in test-time reinforcement learning without access to target ground-truth labels. By emphasizing incorrect majorities and correct minorities during rectifier training, our method improves reward accuracy, which in turn leads to better model performance. Across all evaluated configurations, TRUER improves model performance over naive TTRL and narrows the gap to the Oracle reference trained with ground-truth rewards. This offers a practical path toward adaptation where ground-truth feedback for new problems is costly or infeasible.

AI USE STATEMENT

We used generative AI tools to assist with experimental design, code implementation, and selected aspects of result analysis, as well as manuscript formatting checks, verification of cross-references and bibliographic metadata, and editorial revisions. The authors take full responsibility for the experimental design, code, reported results and their interpretation, and the final manuscript.

REFERENCES

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. arXiv:2407.21787, 2024. URL https://arxiv.org/abs/2407.21787.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, et al. Evaluating Large Language Models Trained on Code. arXiv:2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948, 2025. URL https://arxiv.org/abs/2501.12948.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. The Llama 3 Herd of Models. arXiv:2407.21783, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Jiahui Li, Jianfeng Shan, Wenpei Chen, Shunyu Wu, Jian Lou, Wenjie Feng, Dan Li, and See-Kiong Ng. Exploiting Verification-Generation Gap: Test-Time Reinforcement Learning with Confidence-Conditioned Verification. arXiv:2606.03608, 2026. URL https://arxiv.org/abs/2606.03608.

Siyuan Li, Weiyang Jin, Zedong Wang, Fang Wu, Zicheng Liu, Cheng Tan, and Stan Z. Li. SemiReward: A General Reward Model for Semi-supervised Learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6d1fbfed3f5758c1e5a5a6357a6aa3b1-Abstract-Conference.html.

Teng Pan, Yuchen Yan, Zixuan Wang, Ruiqing Zhang, Guiyang Hou, Wenqi Zhang, Weiming Lu, Jun Xiao, and Yongliang Shen. CoVerRL: Breaking the Consensus Trap in Label-Free Reasoning via Generator-Verifier Co-Evolution. arXiv:2603.17775, 2026. URL https://arxiv.org/abs/2603.17775.

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:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters. arXiv:2408.03314, 2024. URL https://arxiv.org/abs/2408.03314.

Li Wang, Xiaodong Lu, Xiaohan Wang, Yikun Ban, Jiajun Chai, Wei Lin, Tianhao Peng, and Guojun Yin. When Self-Belief Misleads: Active Label Acquisition for Reinforcement Learning with Verifiable Rewards. arXiv:2605.25864, 2026. URL https://arxiv.org/abs/2605.25864.

Ru Wang, Wei Huang, Qi Cao, Yusuke Iwasawa, Yutaka Matsuo, and Jiaxian Guo. Self-Harmony: Learning to Harmonize Self-Supervision and Self-Play in Test-Time Reinforcement Learning. arXiv:2511.01191, 2025. URL https://arxiv.org/abs/2511.01191.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171, 2022. URL https://arxiv.org/abs/2203.11171.

Sitong Wu, Haoru Tan, Xichen Zhang, Bin Xia, Shaofeng Zhang, Xiaojuan Qi, Bei Yu, and Jiaya Jia. Beyond Majority Voting: Self-Reflective Test-Time Reinforcement Learning for LLM Reasoning. Author manuscript, ICML 2026, 2026. URL https://www.cse.cuhk.edu.hk/~byu/papers/C357-ICML2026-SR-TTRL.pdf.

Dong Yan, Jian Liang, Yanbo Wang, Shuo Lu, Ran He, and Tieniu Tan. What If Consensus Lies? Selective-Complementary Reinforcement Learning at Test Time. arXiv:2603.19880, 2026. URL https://arxiv.org/abs/2603.19880.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. arXiv:2409.12122, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 Technical Report. arXiv:2505.09388, 2025a. URL https://arxiv.org/abs/2505.09388.

Shenzhi Yang, Guangcheng Zhu, Xing Zheng, Yingfan Ma, Zhongqi Chen, Bowen Song, Weiqiang Wang, Junbo Zhao, Gang Chen, and Haobo Wang. TraPO: A Semi-Supervised Reinforcement Learning Framework for Boosting LLM Reasoning. arXiv:2512.13106, 2025b. URL https://arxiv.org/abs/2512.13106.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689, 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning. arXiv:2504.16084, 2025. URL https://arxiv.org/abs/2504.16084.

A Experimental Setup and Protocols

A.1 Configurations and Supervision

Table 3 records the nine main-table configurations. The six matched deployment comparisons use the same score-weighted answer aggregation rule with 32 candidates and majority fallback (denoted Weighted in Table 3). Source-label budgets vary across rows.

Table 3: **Configurations in the nine-row main table.** Seeds describe policy training. Rectifier steps identify the frozen checkpoint, not additional policy updates. A dash denotes a configuration outside the matched deployment comparison.

Model	Target	Seed	Updates	Rectifier	Step	Deploy
Math-1.5B	MATH500	42	90	Qwen3-4B	150	Weighted
Qwen3-1.7B	AMC23	42	120	Qwen3-1.7B	198	Weighted
Qwen3-1.7B	AMC23	42	120	Qwen3-4B	25	Weighted
Qwen3-1.7B	AIME24	42	120	Qwen3-1.7B	260	Weighted
Qwen3-1.7B	AIME24	42	120	Qwen3-4B	312	—
Qwen3-4B	AMC23	43	120	Qwen3-4B	175	Weighted
Qwen3-4B	MATH-L5	43	240	Qwen3-4B	50	Weighted
Llama-3.1-8B	MATH500	43	120	Qwen3-4B	175	—
Llama-3.1-8B	AMC23	43	120	Qwen3-4B	175	—

Qwen2.5-Math-1.5B on MATH500. We reserve 130 MATH500 questions for rectifier development and use the remaining 370 for test-time adaptation and evaluation. The eligible source pool contains 3,648 candidates from 124 questions. The selected rectifier consumes 2,400 distinct candidates and generates 21,016 judgment trajectories through update 150. Target questions are excluded from rectifier fitting by ID, normalized text, and the recorded question-family check.

Source candidates were generated by a model after one warm-up round on all MATH500 inputs, without access to any ground-truth answers. Rectifier training and checkpoint selection were then performed exclusively on source data. The models used for evaluation restart from pretrained weights.

AMC23. The configurations using the 1.7B model fit rectifiers on 528 candidates from 20 AMC2024 questions, with ten further AMC2024 questions for source calibration. Test-time adaptation and evaluation use the same 40 AMC23 questions. The source and target files have no overlapping IDs, normalized texts, or digit-template families. The 4B rectifier is frozen at step 25. The main adaptation run with the 4B rectifier uses seed 42; its naive TTRL and Oracle references are shared with the 1.7B-rectifier configuration.

The stable 1.7B rectifier uses the same source data and category weights, learning rate 10^{-6} with ten-step warmup, and KL coefficient 0.02. Its step 198 checkpoint is frozen for test-time adaptation. The model uses seed 42, with historical Naive and Oracle controls; its random stream differs from those controls. This is not an isolated rectifier-size comparison.

Training repetitions. For the Qwen3-1.7B model with the 4B rectifier on AMC23, naive TTRL, TRUER, and Oracle were each independently trained from the same pretrained policy weights for each seed (42, 43, and 44), yielding nine runs evaluated at the final update 120. The rectifier was held fixed across seeds. Table 4 reports all three methods for each seed. TRUER improves Pass@1 over the independently trained naive TTRL run with the same seed by 3.28, 5.78, and 4.84 percentage points, respectively (4.64 points on average). The main table reports seed 42 for consistency with the historical naive TTRL and Oracle controls; this is the most conservative of the three runs in terms of Pass@1 gain.

Table 4: AMC23 training repetitions: Qwen3-1.7B with the fixed 4B rectifier. Pass@1 (%) at update 120. Each row contains independently trained policies using the same training seed. Δ is TRUER minus naive TTRL in percentage points, computed before rounding.

Seed	Naive TTRL	TRUER	Oracle	Δ
42	56.09	59.38	64.61	+3.28
43	55.23	61.02	64.30	+5.78
44	54.84	59.69	65.31	+4.84
Mean	55.39	60.03	64.74	+4.64

AIME24. Both AIME24 configurations use a Qwen3-1.7B model, 30 AIME2025 source questions for rectifier fitting, and all 30 AIME2024 questions for test-time adaptation and evaluation. The 1.7B and 4B rectifiers are frozen at epoch 5 (step 260) and epoch 6 (step 312), respectively. The two configurations share the same Base, naive TTRL, and Oracle references. All policy training runs use seed 42 and 40 epochs (120 updates), with ten questions per update and 32 responses per question. Response generation uses temperature 1, a 2,048-token prompt limit, and a 4,096-token response limit. Main-table reward accuracy uses updates 117–120: out of 1,280 assignments per arm, Naive has 786 correct, and TRUER has 1,111 with the 1.7B rectifier and 1,130 with the 4B rectifier. The latter model's Pass@1 is 238/960, compared with 237/960 for Oracle, yielding 102.3% gap closure; Oracle is an empirical reference rather than a strict performance ceiling.

Additional 4B configurations. The 4B AMC row uses the recorded mixed-100 source rectifier. MATH-L5 uses a separate 600-question target set. The retained summary does not itemize the complete source-label budgets for these two rectifiers; neither inherits the MATH130 or AMC20 budget by assumption.

Llama-3.1-8B-Instruct on MATH500. The Llama configuration uses the fixed MATH500 split of 100 source-fitting questions, 30 source-calibration questions, and 370 target questions. We generate 64 source candidates per question with Llama before fitting the Qwen3-4B rectifier. Rectifier step 175 is selected using source calibration and frozen for test-time adaptation. All three adaptation runs use seed 43, 120 updates over ten epochs, batches of up to 32 questions with 32 responses per question, learning rate 5×10^{-7}, three warmup steps, KL coefficient 0.001, a 4,096-token response limit, and full-parameter training.

All endpoint outputs are scored with the same retained scorer. Evaluation uses four logical shards, eight-question batches, and question-specific seeds. Deployment verification has not been measured.

Llama-3.1-8B-Instruct on AMC23. This configuration reuses the frozen Qwen3-4B step 175 rectifier from the Llama/MATH500 experiment without AMC-specific retraining. Rectifier fitting uses 100 MATH500 source questions, with 64 candidates per question generated by the original Llama model and labeled using source ground truth. All three AMC adaptation runs start from the original Llama-3.1-8B-Instruct model, use seed 43, and complete 120 updates on eight GPUs. Adaptation and endpoint evaluation use all 40 AMC23 questions, with 32 responses per question for the sampled evaluation. These are model endpoint scores; deployment rectifier selection has not been measured for this configuration.

Experimental coverage. Hardware availability and time constraints limited our coverage of the full matrix of language models, rectifiers, tasks, and training seeds. Nevertheless, TRUER achieves higher model Pass@1 and reward accuracy than naive TTRL in all nine configurations reported in Table 2. We view these results, together with the three-seed AMC23 comparison, as supporting the effectiveness of the method within the evaluated settings.

A.2 Metrics and Recorded Protocols

For c_q correct answers among $N = 32$ responses to question q,

$$\widehat{\text{Pass@1}} = \frac{1}{Q}\sum_q \frac{c_q}{32}, \qquad \widehat{\text{Pass@32}} = \frac{1}{Q}\sum_q \mathbb{1}[c_q > 0], \tag{4}$$

$$\widehat{\text{Pass@2}} = \frac{1}{Q}\sum_q \left[1 - \frac{(32-c_q)(31-c_q)}{32\cdot 31}\right]. \tag{5}$$

Majority@32 uses equivalent-answer clusters from the same responses. Unparsed responses remain in the denominator. The main table uses the final model checkpoint, with no deployment verification in the Pass metrics. Gap closure is $(\text{Pass@1}_{\text{TRUER}} - \text{Pass@1}_{\text{Naive}})/(\text{Pass@1}_{\text{Oracle}} - \text{Pass@1}_{\text{Naive}})$, computed before rounding.

Main-table reward accuracy covers each Qwen arm's final four updates: 87–90 for Qwen2.5-Math-1.5B/MATH500, 117–120 for AMC23 and AIME24, and 237–240 for L5. These contain 4,096 responses per Qwen2.5-Math arm and 1,280 per other Qwen arm, including constant-reward groups. The Llama/MATH500 row uses the complete 120-update audit, with 118,400 responses per arm: naive TTRL has 69,295 TP, 27,737 FP, 20,284 TN, and 1,084 FN; TRUER has 75,309 TP, 2,208 FP, 38,500 TN, and 2,383 FN. Reward accuracy is 75.66% and 96.12%, respectively. The Llama/AMC23 row also uses all 120 updates, with 38,400 responses per arm: naive TTRL has 15,710 TP, 14,571 FP, 7,898 TN, and 221 FN; TRUER has 15,215 TP, 1,085 FP, 19,144 TN, and 2,956 FN, giving reward accuracies of 61.48% and 89.48%, respectively. These counts come from each arm's own trajectory; fixed-pool audits instead compare labeling rules on identical saved responses.

The Qwen2.5-Math-1.5B settings appear in Table 5. AMC adaptation runs use 120 updates, ten questions per update, 32 responses per question, a 4,096-token response cap, and a 5,120-token context. The rectifier uses greedy decoding with a 4,096-token output cap and 8,192-token context. The L5 endpoint is update 240. Reward selection never receives target gold; Oracle uses it explicitly.

Table 5: Qwen2.5-Math-1.5B/MATH500 training settings.

Setting	Value
Model	Qwen2.5-Math-1.5B, original pretrained weights
Policy trainable parameters	All parameters; four-GPU FSDP2
Policy optimizer	AdamW; weight decay 0.01; gradient clipping 1.0
Policy learning rate	Peak 5×10^{-7}; three warmup steps; cosine to step 90
Reference policy / KL	Frozen initial policy; separate KL loss coefficient 0.001
Training batch / responses	32 questions / 32 responses per question
Generation settings	Temperature 1; top-p 1; top-k disabled
Generation lengths	Prompt 1,024; response 3,072; context 4,096 tokens
Rectifier	Qwen3-4B; LoRA (Hu et al., 2021) rank 64, scaling alpha 128
Rectifier optimizer	Learning rate 5×10^{-6}; weight decay 0.01
Rectifier training groups	16 candidate prompts per batch; initially 8 judgments per candidate
Rectifier extra sampling	Same-policy expansion to 16/32 for homogeneous-reward groups that are not all successful
Rectifier PPO / KL	Clip 0.2; KL coefficient 0.01
Rectifier decoding	Training temperature 1; native reasoning; output limit 4,096, context 8,192
Rectifier deployment	Frozen step 150; greedy reasoning; score ≥ 5 accepts
Evaluation	Sampled $N = 32$, temperature 1; greedy separately at temperature 0
Evaluation scheduling	Four-question batches; fixed question-specific hash seeds

MATH TTRL and TRUER resume at step 30 with optimizer, scheduler, and random states restored; rank1-only and Oracle run continuously. TTRL resolves top-count ties by sample order, whereas rectifier policies abstain. The rank1-only/TRUER ablation shares the tie rule.

Scoring and uncertainty. We retain the frozen mathematical scorer across arms. Known edge cases involving intervals, units, equations, and numeric bases are not silently corrected in the main results. An audit affected sixteen rectifier candidate labels, including four training candidates; the selected MATH rectifier was not retrained after this audit. Reviewed examples exclude known scoring errors. Paired bootstrap intervals resample question IDs 10,000 times and do not include training-seed or checkpoint-selection uncertainty.

A.3 Rectifier Training and Selection Rules

The rectifier receives the problem, candidate solution, and extracted answer. Its prompt asks for an independent check and a final JSON judgment containing a reason and score. It neither executes candidate code nor receives gold answers or category labels. Scores are learned ratings rather than calibrated probabilities. Valid outputs use Equation 2; malformed or truncated outputs receive -1.2.

For the rectifier used with Qwen2.5-Math-1.5B on MATH500, category k with n_k candidates and priority α_k has weight

$$w_k = \frac{M\alpha_k}{n_k \sum_j \alpha_j}, \qquad M = 3{,}648. \tag{6}$$

The weight multiplies the policy loss after advantage normalization. Table 6 describes the eligible candidate pool.

Table 6: **Rectifier supervision for Qwen2.5-Math-1.5B/MATH500.** Correctness labels come from source ground truth; category membership is not provided as input.

Category	Eligible candidates	Priority	Loss weight
Majority-correct	1,621	1	0.2813
Majority-wrong	440	3	3.1091
Minority-correct	153	3	8.9412
Minority-wrong	1,434	1	0.3180

During test-time adaptation, rank1 is checked using its first eligible representative in sampling order. Eligibility uses parsing and length. A valid score $s \geq 5$ accepts the answer; $s < 5$ permits a rank2 check if there is no boundary tie. Missing representatives, unresolved judgments, and rejected alternatives yield abstention. Rank1-only removes the second check. Abstained groups have zero outcome advantage; the separate KL term remains active.

B Additional Results and Learning Dynamics

Original figure legends labeled "Ours" or "Full cascade" refer to TRUER.

B.1 MATH500 Learning Curves and Checkpoints

The MATH results below retain all recorded checkpoints and the rank1 ablation. The extra update 110 continuation has a longer training budget and a changed learning-rate phase; it is supplementary to the update 90 main comparison.

Figure 6 compares the recorded learning curves, and Figure 7 provides paired uncertainty estimates. Table 7 lists the checkpoint scores; the rank1-only contrasts are reported in Appendix C.3.

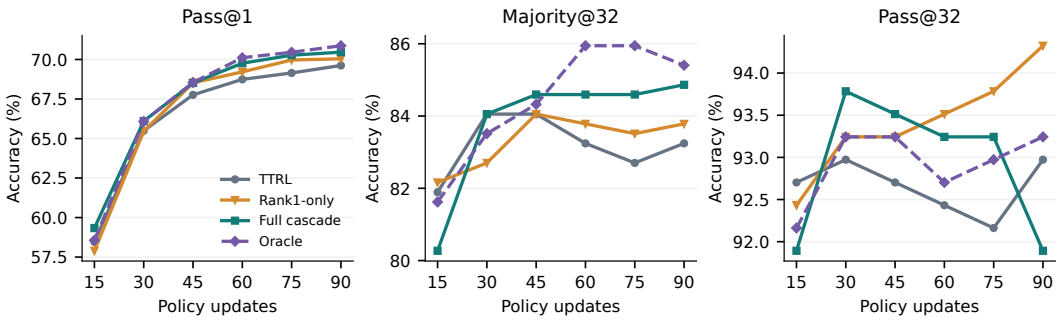

Figure 6: **Reward intervention and policy learning.** TRUER moves Pass@1 and Majority@32 toward the ground-truth-reward reference, while Pass@32 shows a different ordering. Each line is one training seed. The figure shows all recorded checkpoints and does not select the best intermediate result.

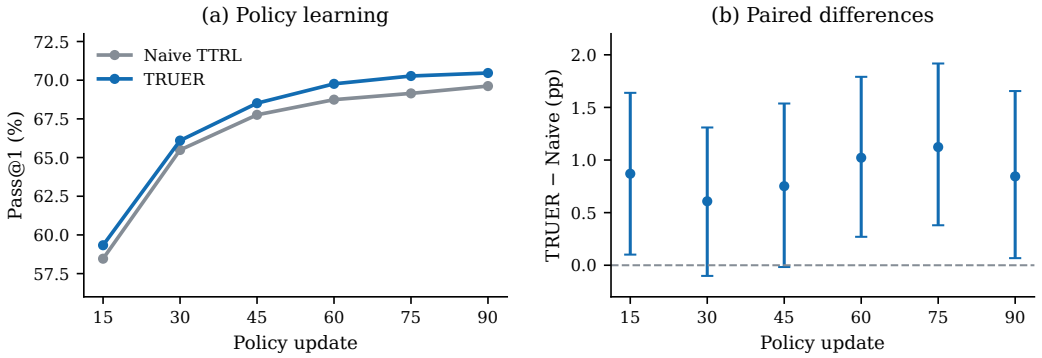

Figure 7: **Paired uncertainty in the MATH500 learning curves.** Left: model Pass@1. Right: TRUER minus TTRL with pointwise 95% question-bootstrap intervals, conditional on the saved samples and one training seed.

Table 7: All recorded model checkpoints. Percentages, frozen scorer. No best-checkpoint selection is used for the main endpoint.

Step	Method	Greedy	Pass@1	Pass@2	Majority@32	Pass@32
15	TTRL	73.51	58.46	72.43	81.89	92.70
15	Rank1-only	72.70	57.90	72.33	82.16	92.43
15	TRUER	72.43	59.33	73.07	80.27	91.89
15	Oracle	72.16	58.56	72.49	81.62	92.16
30	TTRL	72.43	65.49	76.99	84.05	92.97
30	Rank1-only	72.97	65.46	76.77	82.70	93.24
30	TRUER	74.59	66.10	77.16	84.05	93.78
30	Oracle	74.59	66.08	77.35	83.51	93.24
45	TTRL	75.68	67.76	78.22	84.05	92.70
45	Rank1-only	76.22	68.52	78.75	84.05	93.24
45	TRUER	75.68	68.51	78.87	84.59	93.51
45	Oracle	76.76	68.55	79.07	84.32	93.24
60	TTRL	77.30	68.74	78.81	83.24	92.43
60	Rank1-only	75.68	69.21	79.04	83.78	93.51
60	TRUER	75.95	69.76	79.44	84.59	93.24
60	Oracle	78.11	70.11	79.99	85.95	92.70
75	TTRL	78.92	69.15	78.68	82.70	92.16
75	Rank1-only	77.57	69.97	79.27	83.51	93.78
75	TRUER	75.95	70.27	79.73	84.59	93.24
75	Oracle	78.65	70.46	80.03	85.95	92.97
90	TTRL	76.22	69.62	79.33	83.24	92.97
90	Rank1-only	76.22	70.04	79.51	83.78	94.32
90	TRUER	76.49	70.46	79.76	84.86	91.89
90	Oracle	77.30	70.87	80.32	85.41	93.24

B.2 AMC23 Learning Curves and Endpoints

The following AMC figures use the seed 42 run with the 1.7B model and 4B rectifier reported in the main table. This run is distinct from the stable 1.7B-rectifier configuration. Only TRUER intermediate sampled checkpoints were retained; Naive and Oracle lines mark their endpoints.

Figure 8 shows the recorded learning curves; Table 8 reports the endpoint metrics.

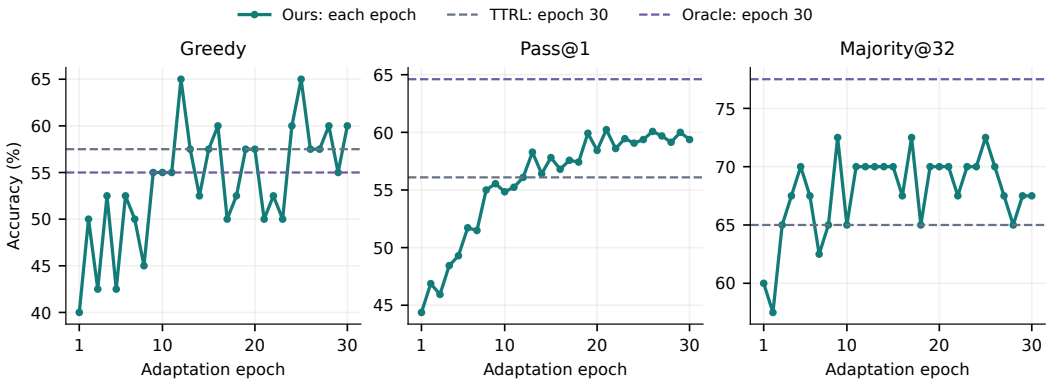

Figure 8: **All 30 AMC23 TRUER checkpoints.** Solid curves are separately completed, matched evaluations of each checkpoint without rectifier-based selection. Dashed lines mark the final epoch 30 TTRL and Oracle scores; their intermediate sampled checkpoints were not retained. Greedy uses the same complete evaluation pipeline rather than the separate inline training validation.

Table 8: AMC23 endpoints: 1.7B model, 4B rectifier, seed 42 (%).

Method	Greedy	Pass@1	Pass@2	Majority@32	Pass@32
Base	45.00	46.09	59.97	67.50	90.00
TTRL	57.50	56.09	61.67	65.00	77.50
TRUER	60.00	59.38	66.67	67.50	92.50
Oracle	55.00	64.61	72.91	77.50	92.50

B.3 Answer-Distribution Changes

The base model already samples a correct answer for 330 of the 370 MATH questions. Most additional correct samples after adaptation come from this already-covered set. Figure 9 tracks each question's correct-answer count; no questions are selected by improvement. TRUER newly covers fifteen questions and loses five, alongside a larger increase in the frequency of correct answers. These are finite-sample coverage observations.

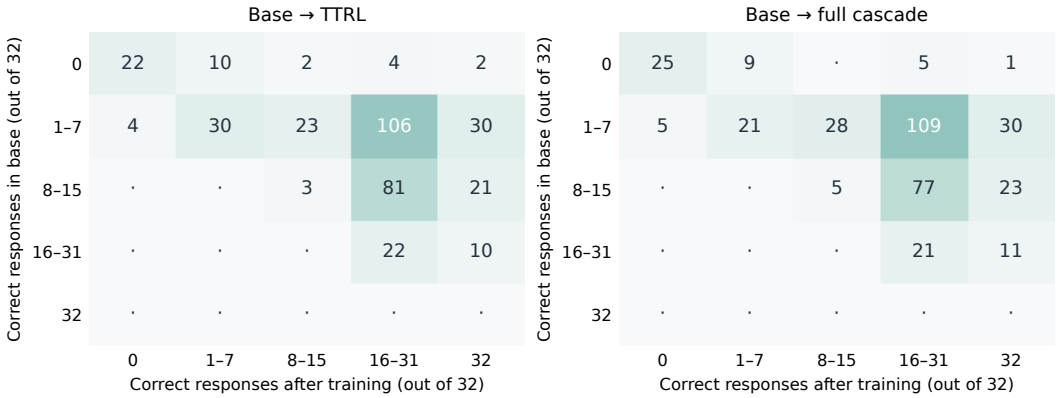

Figure 9: **Where the same questions move after adaptation.** Rows are base correct-response counts; columns are counts at update 90. Both panels contain all 370 questions and use the same bins and color scale. Movement to the right indicates more frequent correct answers.

Table 9 records parsing rates and generation lengths for the same endpoint comparison.

Table 9: Parsing and generation lengths under the base comparison. All scores retain the full denominator of 11,840 responses; unparsed responses are not discarded.

Model	Unparsed	Length-limit hits	Mean output tokens
Base	32.97%	11.25%	1,102.7
TTRL90	1.78%	2.58%	682.6
TRUER90	1.98%	2.52%	684.1

C Reward Analysis and Ablations

C.1 Fixed-Pool Reward Correction

The fixed MATH TRUER trajectory contains 92,160 responses and 2,880 question visits. Majority labeling produces 4,012 FP and 1,088 FN rewards; rectification produces 1,081 FP and 2,284 FN rewards. Reward accuracy rises from 94.47% to 96.35%. TRUER retains 55,039 originally correct positive rewards, removes 1,557, and recovers 361, leaving 55,400 in total. Active groups fall from 2,557 to 2,105.

At the question-visit level, 48 wrong labels become correct, 357 become abstentions, and 99 correct majorities are withheld. The rank2 branch accepts 48 correct and 25 wrong alternatives. Figure 10 relates these decisions to retained advantage mass and candidate availability. Counts are visits, not distinct questions.

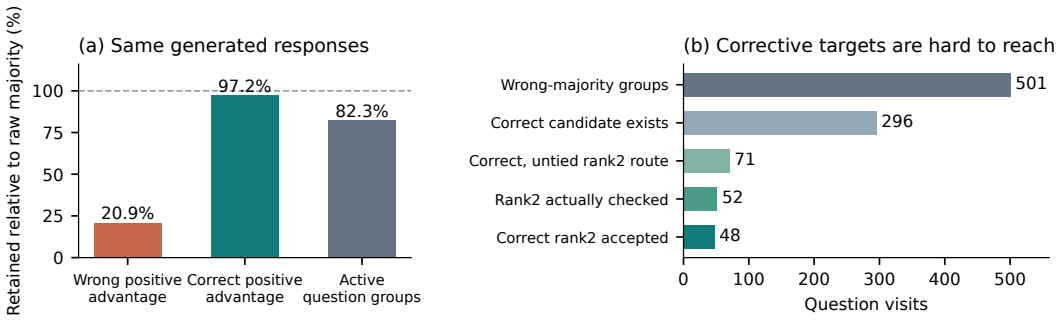

Figure 10: **Cleaner rewards do not automatically supply more corrective signal.** Both panels use TRUER's fixed training response pool. Left: retained advantage mass and active groups relative to raw majority labeling. Right: the nested conditions needed to rescue a correct rank2. Counts are question visits, not unique questions.

Table 10 separately reports the full-run reward audit on each method's own responses.

Table 10: Training reward audit over all 90 updates, evaluated on each arm's own responses. For an isolated labeling comparison, use the shared-pool numbers in the main text.

Quantity	TTRL	TRUER
TP	56,183	55,400
FP	4,363	1,081
TN	30,498	33,395
FN	1,116	2,284
Overall reward accuracy	94.05%	96.35%
FPR	12.52%	3.14%
Positive-reward precision	92.79%	98.09%
Correct-response reward recall	98.05%	96.04%
Accepted question-group coverage	100.00%	84.17%

C.2 Comparison with Vote Filters

We replay vote-count and vote-margin thresholds on the same response pool (Table 11 and Figure 11). At comparable FP removal, the rectifier retains more correct rewards and active groups than these frequency-only filters. The margin rule requires discarding about 2.7 times as many correct positive rewards to match TRUER's FP count. At aggressive thresholds it retains less wrong-positive advantage mass, but also loses more correct-positive mass.

Table 11: Label-free confidence filters on the frozen pool of Section 5.3. Only vote counts enter the selection. Precision is $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$; the last two columns are relative to unfiltered TTRL majority. The learned cascade removes 73.1% of false positives while keeping 97.9% of correct positive rewards and 82.3% of active groups.

Policy	TP	FP	Active groups	Precision (%)	TP kept (%)	FP removed (%)
TTRL (no filter)	56,596	4,012	2,557	93.38	100.0	0.0
Vote margin δ=8	54,133	1,299	1,815	97.66	95.6	67.6
Vote margin δ=10	53,143	1,014	1,714	98.13	93.9	74.7
Vote margin δ=12	52,110	928	1,636	98.25	92.1	76.9
Absolute support ≥ 24	43,692	496	1,179	98.88	77.2	87.6
TRUER (ours)	55,400	1,081	2,105	98.09	97.9	73.1

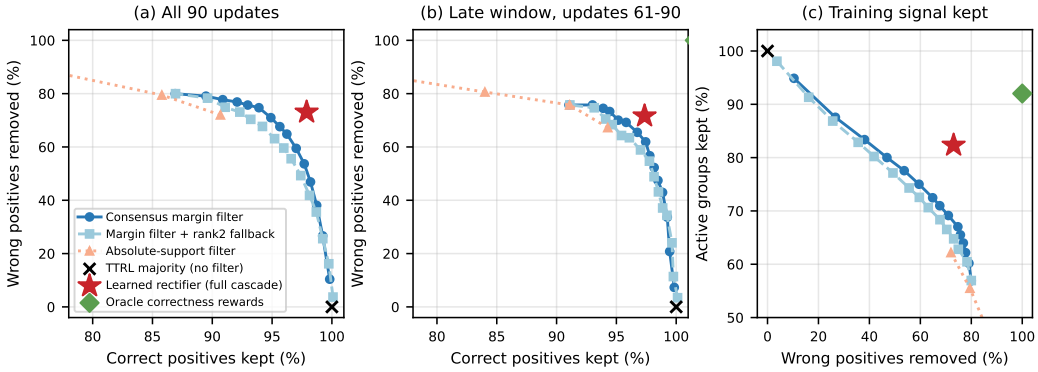

Figure 11: **Label-free confidence filters against the learned rectifier.** Each curve sweeps one filter family over its threshold. At comparable FP removal, TRUER retains more correct rewards and active groups. Panel (b) restricts the pool to updates 61–90, where false positives dominate reward errors.

C.3 RANK1-ONLY ABLATION

Table 12 reports paired endpoint contrasts between TRUER, rank1-only, and naive TTRL. The checkpoint scores are listed in Table 7.

Table 12: Paired 90-step contrasts involving rank1-only. Differences and intervals are percentage points. These were recomputed from the saved question-level scores after the rank1 run completed.

Contrast	Metric	Difference	95% interval
TRUER − rank1	Greedy	+0.27	[-2.43, +2.97]
TRUER − rank1	Pass@1	+0.42	[-0.37, +1.28]
TRUER − rank1	Pass@2	+0.25	[-0.54, +1.11]
TRUER − rank1	Majority@32	+1.08	[-1.35, +3.51]
TRUER − rank1	Pass@32	-2.43	[-4.32, -0.81]
Rank1 − TTRL	Greedy	+0.00	[-2.70, +2.70]
Rank1 − TTRL	Pass@1	+0.42	[-0.37, +1.17]
Rank1 − TTRL	Pass@2	+0.17	[-0.68, +1.00]
Rank1 − TTRL	Majority@32	+0.54	[-1.89, +2.97]
Rank1 − TTRL	Pass@32	+1.35	[-0.54, +3.24]

C.4 Effect of Rectifier Training

We compare the original Qwen3-4B and the trained rectifier on identical AMC23 candidate pools, using the same tokenized prompts, greedy decoding, parser, eligibility rules, and score threshold. The pools come from the base model, naive TTRL, and Oracle. Invalid outputs count as errors. Table 13 reports the numerical comparison, visualized in Figure 12.

Table 13: **Original versus trained rectifier on identical AMC23 responses.** Each entry is original→trained, in percent. Accuracy includes invalid outputs as errors. Wrong-answer acceptance is conditional on a wrong candidate; correct-answer withholding includes explicit rejection and invalid outputs on correct candidates. The rectifier was trained on disjoint AMC2024 questions.

Response source	Accuracy ↑	Wrong accepted ↓	Correct withheld ↓	Invalid ↓
Base model	74.39 → 88.05	38.47 → 14.80	7.65 → 8.16	3.09 → 0.33
TTRL model	71.85 → 86.52	52.11 → 18.35	4.89 → 9.64	3.97 → 0.16
Oracle model	82.89 → 90.79	48.31 → 16.92	3.19 → 5.89	1.84 → 0.26

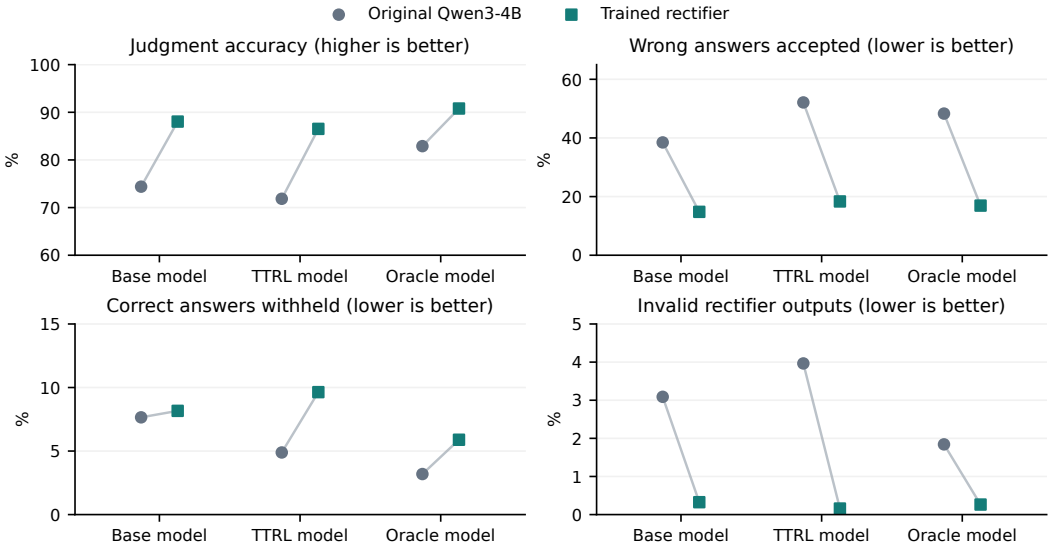

Figure 12: **Task-specific training improves the 4B rectifier's judgment tradeoff.** Original and trained models judge the same candidates. Training reduces wrong-answer acceptance but also withholds more correct answers.

Training improves judgment accuracy and reduces wrong-answer acceptance across these pools, with some increase in correct-answer withholding. Paired question-bootstrap intervals exclude zero for accuracy improvements and wrong-answer acceptance reductions. This control evaluates judgments; it does not isolate category weighting or replace a matched adaptation run using the untrained rectifier.

Adaptation with an untrained rectifier. Table 14 compares adaptation using the original pretrained rectifier and its trained counterpart. All three reported configurations start from the original Qwen3-1.7B policy and use seed 42, 120 policy updates, and the original endpoint evaluation with 32 responses per question. The trained-rectifier results use the original endpoints rather than the budget-control continuations. The trained rectifier yields higher final Pass@1 in each comparison; the untrained 1.7B rectifier on AIME24 also underperforms naive TTRL.

Table 14: **Adaptation with and without rectifier training.** Final model Pass@1 (%) without rectifier-based answer selection. All policies are Qwen3-1.7B; untrained denotes the original pre-trained rectifier without task-specific training.

Target	Rectifier	Naive TTRL	Untrained	Trained
AMC23	Qwen3-4B	56.09	59.22	**59.38**
AIME24	Qwen3-1.7B	20.21	18.96	**21.46**
AIME24	Qwen3-4B	20.21	23.02	**24.79**

These comparisons have execution differences: the AMC23/4B and AIME24/4B untrained runs use policy data parallelism of four rather than two, while AIME24/1.7B changes from RTX 5090 with vLLM 0.23 to A800 with vLLM 0.25.1. They therefore do not isolate rectifier training under identical hardware and runtime settings.

D Case Studies

We select the successful example by the largest endpoint correct-sample gap among reviewed questions where naive TTRL is unanimously wrong and TRUER has at least 75% correct responses. The failure example has the largest combined wrong-mode share among reviewed cases where both models have at least 75% wrong consensus. Ties use question IDs. Known scorer errors and mixed-truth clusters are excluded.

Recovering a correct minority. For MATH algebra/2023, the question asks for the smallest integer in the domain of $\sqrt{x-1}/(x-2)$. The constraints are $x \geq 1$ and $x \neq 2$, so the answer is 1; a common mistake gives 3. TRUER rejects 3 and selects the minority answer 1 at updates 10, 13, and 34. Later visits have a correct majority. Endpoint samples contain 24 correct answers for TRUER, while naive TTRL answers 3 in all 32 samples.

Filtering without recovery. For intermediate algebra/1279, summing the roots of $\sum_{j=1}^{4} j/(x^2 - j) = 2010x - 4$ gives $2/1005$ by Vieta's formula after clearing denominators. Both models commonly answer zero. TRUER removes the wrong majority reward on all eight visits but abstains each time; the few correct candidates are blocked by rank2 tie rules. Neither model produces a correct endpoint sample (Figure 13). Cleaner rewards can prevent an erroneous update without producing a corrective one.

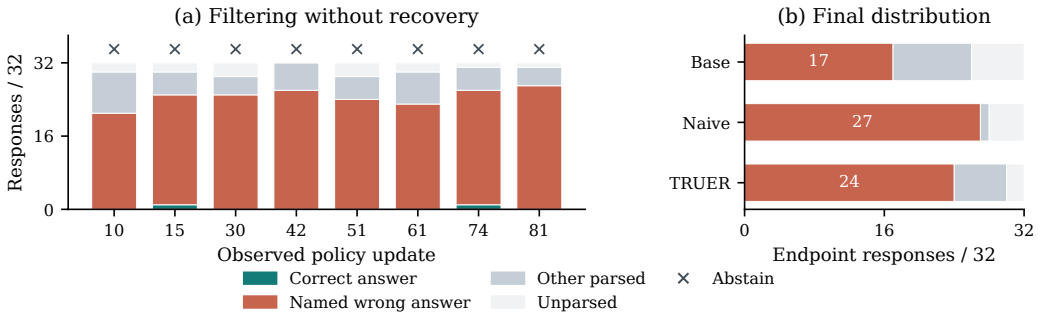

Figure 13: **Filtering without recovery.** (a) Answer distributions at all eight recorded TRUER training visits; crosses mark abstention. (b) Separately sampled endpoint responses for Base, naive TTRL, and TRUER. The named wrong answer is zero. Rejecting this answer supplies no replacement target, and the model remains incorrect.

An incorrect fallback. At update 64 on intermediate algebra/1849, the rectifier rejects rank1 answer 2 but accepts rank2 answer 1 with score 9. The true answer is 501, absent from the candidate

pool. The judgment misses a family of negative integer cases. Rejecting one wrong answer therefore does not certify its alternative.

E Deployment Selection and Cost

E.1 Shared Selection Rule

The current deployment comparison uses one score-weighted answer aggregation rule for all six configurations. We retain each final model's original pool of 32 responses and apply the configuration's frozen rectifier to both naive TTRL and TRUER. A response is eligible when its final answer can be parsed and clustered, the model response is not truncated, and the rectifier prompt fits the 4,096-token input limit. Rectifier judgments use greedy decoding, a 4,096-token output cap, and an 8,192-token context. A valid integer score of at least 5 passes; malformed or truncated judgments do not pass.

For each answer a, let $\mathcal{V}_a$ contain the eligible candidates with that answer and a valid rectifier score s_i, and let $n_a = |\mathcal{V}_a|$. We compute

$$S(a) = \frac{\sum_{i \in \mathcal{V}_a} \max(s_i - 4, 0)}{\sqrt{n_a}}, \qquad n_a > 0. \tag{7}$$

Valid rejected judgments contribute zero to the numerator but remain in n_a; invalid judgments are excluded from both. The weights are used for ranking and are not calibrated probabilities. We select the answer with the largest positive support. Ties are resolved by the highest passing score in the answer cluster, then its original total candidate count, and finally the earliest index among its highest-scoring passing candidates. If no candidate passes, we return the original majority answer. Every question remains in the denominator.

Selector choice and scope. We selected this shared aggregation rule after comparing alternative selectors on the target response pools. The resulting deployment comparisons are exploratory and do not constitute independent validation of the selector choice. Candidate selection itself receives no target labels. The model checkpoints, rectifiers, candidate pools, eligibility criteria, and answer scorers remain fixed; this replay changes only how the saved judgments are aggregated. In particular, the AMC23 configuration with a 1.7B model and 4B rectifier uses the original seed 42 candidate pools and judgments.

Table 15 reports the paired naive TTRL and TRUER results. The shared aggregation rule improves over each TRUER model's own majority voting in all six configurations. With matched rectifier postprocessing, TRUER is higher than naive TTRL in three configurations, tied in one, and lower in two. Historical selection controls below retain their original rules and results.

Table 15: **Deployment under one shared selection rule (%).** Parentheses identify the rectifier size. Both adaptation methods use the same frozen rectifier within each row. "+Rectifier" applies Equation 7 to the corresponding model's fixed 32-response pool, with majority fallback.

		Naive TTRL		TRUER	
Model / target (rectifier)	N	Majority	+Rectifier	Majority	+Rectifier
Math-1.5B / MATH500 (4B)	370	83.24	88.11	84.86	86.76
Qwen3-1.7B / AMC23 (1.7B)	40	65.00	65.00	67.50	70.00
Qwen3-1.7B / AMC23 (4B)	40	65.00	70.00	67.50	70.00
Qwen3-1.7B / AIME24 (1.7B)	30	26.67	30.00	26.67	33.33
Qwen3-4B / AMC23 (4B)	40	82.50	85.00	90.00	95.00
Qwen3-4B / MATH-L5 (4B)	600	84.83	87.17	86.00	86.17

E.2 Historical Selection Controls

Table 16 uses the same AMC1.7B/4B-rectifier seed 42 pools as the main table and the deployment comparison. It additionally tests top2 with one representative per cluster and top2 with all eligible responses in those clusters. These policies and the All32 selector rank passing candidates by score,

cluster size, and sample order. None of the historical AMC selection gains establishes a statistically significant improvement under the recorded paired tests.

Table 16: **Historical AMC23 final answer-selection accuracy (%).** Every row uses the same 32 candidates for each model, and full question coverage. Changes do not alter the models' sampled Pass@1 or Pass@32.

Selection rule	Base	TTRL	TRUER	Oracle
Original majority	67.50	65.00	67.50	77.50
Sequential rank1-to-rank2	72.50	67.50	67.50	80.00
Top2: one representative	72.50	67.50	65.00	80.00
Top2: all candidates	70.00	65.00	70.00	80.00
All32: all candidates	70.00	65.00	75.00	80.00

Table 17 reports the historical MATH fixed-pool selection controls, including the additional continuation described in Appendix B.

Table 17: Historical MATH fixed-pool selection controls (percent). Oracle receives matched post-processing here.

Selection rule	TRUER90	TRUER110	Oracle90
Original majority	84.86	85.14	85.41
Rank1-to-rank2 cascade	85.68	86.22	86.22
Top2, score first	85.68	85.95	86.76
Top3, score first	85.68	85.68	87.84

E.3 Inference Cost

The weighted aggregation rule reuses the same at most 32 candidate judgments per question as All32. No new judgments are generated for this replay. Table 18 records the judgments and rectifier output tokens for the current paired evaluations; these totals exclude prompt tokens and the unchanged response generation.

Table 18: **Rectifier inference cost for the unified deployment comparison.** Counts cover all eligible responses in each fixed pool.

	Naive TTRL		TRUER	
Model / target (rectifier)	Judgments	Output tokens	Judgments	Output tokens
Math-1.5B / MATH500 (4B)	11,463	7,409,697	11,457	7,382,694
Qwen3-1.7B / AMC23 (1.7B)	1,261	1,501,618	1,207	1,366,736
Qwen3-1.7B / AMC23 (4B)	1,261	1,110,284	1,173	939,509
Qwen3-1.7B / AIME24 (1.7B)	864	208,525	768	169,730
Qwen3-4B / AMC23 (4B)	1,194	936,227	1,170	925,414
Qwen3-4B / MATH-L5 (4B)	17,801	11,997,996	17,359	8,656,022

On historical AMC TRUER candidates, top2 representatives require 67 judgments and 58,074 rectifier output tokens, while All32 requires 1,173 judgments and 939,509 output tokens. These counts exclude input tokens and response generation. Figure 14 shows the recorded tradeoff.

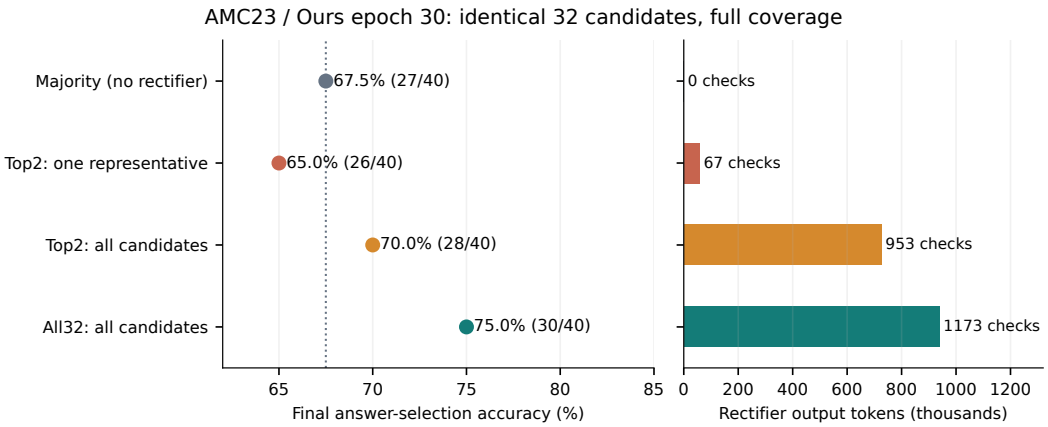

Figure 14: **Selection quality and additional rectifier cost on AMC23.** The same frozen rectifier scores TRUER epoch 30 candidates. Output-token counts are sums from the cached judgments, excluding rectifier input tokens and the unchanged response generation. Calls count candidate judgments. All policies retain all 40 questions.

F Training-Budget Controls

We conduct two independent continuation controls on configurations from the main table. The GPU-hour control extends naive TTRL and compares it with the original TRUER checkpoint. The active-group control extends TRUER and compares it with the original naive TTRL checkpoint; we also report its change from the original TRUER endpoint. The two extended endpoints do not simultaneously match computation and effective training exposure.

Training overhead. The recorded MATH policy updates take approximately 219.3 seconds for TRUER versus 177.2 for naive TTRL, with a dedicated rectifier GPU; this excludes rectifier training. Matching policy updates is not total-compute matching.

F.1 GPU-Hour-Matched Naive TTRL

Each run starts from the original naive TTRL endpoint and continues until its cumulative adaptation-stage GPU-hours approximately match those of the original TRUER run. The stopping budget is fixed before endpoint evaluation. Table 19 reports all five configurations. The final naive budgets are 1.1–1.4% above the corresponding TRUER budgets, reflecting the budget margin and stopping at complete updates. Additional naive training improves Pass@1 in three configurations and reduces it in two; the original TRUER checkpoints remain higher in all five comparisons.

Table 19: **GPU-hour-matched naive continuation.** N_0 and T_0 are the original naive TTRL and TRUER endpoints; N_+ is the extended naive TTRL endpoint. GPU-hours cover adaptation and online rectifier inference, excluding rectifier pretraining, saving, export, and evaluation.

		Pass@1 (%)			GPU-hours		
Model / task	Seed	N_0	N_+	T_0	N_0	N_+	T_0
Qwen3-1.7B / AMC23	42	56.09	59.14	59.38	9.78	27.65	27.31
Qwen3-1.7B / AIME24	42	20.21	23.75	24.79	19.70	39.39	38.96
Qwen3-4B / AMC23	43	79.69	78.52	82.34	27.10	30.64	30.21
Qwen3-4B / MATH-L5	43	79.82	80.36	80.68	56.45	68.75	67.84
Llama-3.1-8B / AMC23	43	49.45	47.50	60.23	29.26	63.69	62.97

F.2 ACTIVE-GROUP-MATCHED TRUER

An active group contains 32 responses with nonconstant binary rewards and hence nonzero group-relative outcome advantages. We extend each original TRUER checkpoint until its cumulative active-group count reaches that of the original naive TTRL run, stopping at the first complete update that crosses the target. This matches effective group exposure, not optimizer-step counts or accumulated parameter-gradient norms. Counts are independently checked against the saved per-response rewards at every update. Table 20 reports the active-group-matched results.

Table 20: **Active-group-matched TRUER continuation.** T_+ is the extended TRUER endpoint. Group counts include the original training; the target is the original naive TTRL count. GPU-hours are cumulative. Seeds and original scores are given in Table 19.

	Active groups					
Model / task	T_0	Target	T_+	Added steps	Pass@1 (%)	GPU-hours
Qwen3-1.7B / AMC23	636	930	935	65	61.02	43.46
Qwen3-1.7B / AIME24	530	1,179	1,180	144	27.92	121.80
Qwen3-4B / AMC23	569	651	653	21	81.72	38.00
Qwen3-4B / MATH-L5	1,094	1,197	1,197	24	80.74	76.15
Llama-3.1-8B / AMC23	415	630	630	139	64.53	140.13†

†The Llama GPU-hour entry is a lower bound; the recorded cumulative cost is 140.13–140.60 GPU-hours, including discarded computation after checkpoint recovery. Active-group counts include only the retained training trajectory.

All five active-group-matched endpoints remain above the original naive TTRL scores. Relative to the original TRUER checkpoints, Pass@1 improves in four configurations, with gains of 3.13 percentage points on Qwen3-1.7B/AIME24 and 4.30 points on Llama/AMC23. Qwen3-4B/AMC23 decreases by 0.63 points, while Qwen3-4B/MATH-L5 changes little ($+0.06$ points). Thus, additional effective training benefits most configurations, but does not uniformly improve TRUER.

F.3 CONTINUATION AND EVALUATION PROTOCOLS

Both controls restart Adam from the corresponding final model weights while retaining the original base reference model. The TRUER continuations retain the frozen rectifier. Budgets determine stopping independently of evaluation scores. Endpoint evaluation retains each original configuration's questions, scoring rules, and full 32-sample plus greedy protocol. GPU-hours count active training time multiplied by the participating physical GPUs, including online rectifier services and counting shared GPUs only once. They exclude rectifier pretraining, checkpoint saving, export, and endpoint evaluation. Hardware allocations and parallel topologies differ in some continuations, so these are approximate adaptation-compute controls rather than hardware-normalized FLOP comparisons or from-scratch runs under a common total budget.

These controls use one training seed per configuration. Conditional question-bootstrap 95% intervals for the TRUER-minus-extended-naive Pass@1 difference are $[0.94, 7.03]$ points on Qwen3-4B/AMC23, $[-10.23, 11.17]$ on Qwen3-1.7B/AMC23, and $[-1.88, 27.81]$ on Llama/AMC23. For extended-minus-original TRUER, the corresponding available intervals are $[-0.55, 3.67]$ on Qwen3-1.7B/AMC23 and $[-2.19, 1.09]$ on Qwen3-4B/AMC23. These intervals condition on the retained runs and do not include training-seed variation.

G REPRODUCIBILITY

Figures and tables use retained question-level scores, reward audits, and response histories. Maintable data, reward-window counts, figure inputs, and source hashes are preserved with the analysis. Historical configurations and exploratory selection comparisons are identified separately. The continuation controls additionally retain budget ledgers and complete endpoint evaluations.