# CRITIQUE TO VERIFY: ACCURATE AND HONEST TEST-TIME SCALING WITH RL-TRAINED VERIFIERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Sampling multiple solutions improves mathematical reasoning only when an aggregation rule can identify useful answers among them. Majority voting favors frequent answers, while a learned verifier must distinguish correctness from agreement. We introduce **Mirror-Critique**, a framework for training generative verifiers from reference-guided critiques. An open instruction-tuned model contrasts solver trajectories with reference solutions, and rejection sampling retains critiques whose final judgments match the candidates' verified answer labels. These critiques provide a supervised cold start before reinforcement learning with verifiable rewards on balanced correct and incorrect candidates. At test time, the resulting **Mirror-Verifier** generates multiple critiques per solution and uses their judgments for weighted voting or selective abstention. Across five mathematical reasoning benchmarks, the reported aggregate accuracy increases from majority voting's 55.7% to 58.2% for the 1.5B solver and from 61.8% to 63.0% for the 7B solver. Under a utility that rewards correct answers, penalizes incorrect answers, and assigns zero to abstention, Mirror-Verifier also improves the reported honesty score. The results support critique-based initialization followed by verifier reinforcement learning as a means of improving both answer selection and selective prediction.

## 1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) trains reasoning models using automatically checked outcomes. In mathematical problem solving, the final answer supplies a convenient reward even when intermediate reasoning is difficult to annotate. At inference time, these solvers can generate several candidate solutions rather than commit to a single trajectory. The resulting gains depend on two distinct abilities: producing a correct candidate and recognizing it among the alternatives. Improving the first does not automatically solve the second.

Majority voting is a strong, inexpensive aggregation rule because repeated answers often indicate a stable solution. However, it cannot distinguish repeated mistakes from repeated correct reasoning. If a correct answer occurs in only a few samples, a larger sample set can reinforce the wrong majority rather than reveal the correct minority. A verifier offers a different source of evidence by examining each candidate's reasoning. Its usefulness depends on whether its scores contain information beyond answer frequency.

Training such a verifier from binary correctness labels is not straightforward. An outcome label indicates whether the final answer matches a reference, but it does not explain which calculation or inference failed. A generative verifier must learn both to inspect a solution and to express a judgment. Reinforcement learning can reward the judgment, yet the reward alone does not supply an example of how to perform the inspection. Reference-guided critique generation provides a practical way to introduce this behavior before optimizing it with RLVR.

We propose **Mirror-Critique**, which constructs critique supervision by contrasting model-generated solutions with reference solutions. Solver trajectories collected during RLVR provide candidates spanning different error patterns. An open instruction-tuned model produces a critique for each candidate, with access to the reference during synthesis. We retain critiques whose final True/False judgments agree with the candidate's automatically checked answer. The resulting data initializes a verifier through supervised fine-tuning (SFT), after which reinforcement learning improves its



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.

judgment accuracy using balanced positive and negative examples. References are used to construct training supervision, not supplied to the verifier at inference time.

The trained **Mirror-Verifier** samples several critiques of each candidate. The fraction of positive judgments becomes a verification score, which supports two decisions. Weighted voting combines this score with answer frequency to select an answer. A separate threshold on the selected answer's average verification score allows the system to abstain. These decisions need not be conflated: a candidate can win a relative comparison without having enough absolute support to justify answering.

We evaluate the framework with Qwen2.5-Math solvers at 1.5B and 7B scales on five mathematical reasoning benchmarks. Mirror-Verifier improves aggregate accuracy over majority voting and the evaluated learned-verifier baselines. We also evaluate selective prediction with an *honesty score*: correct answers receive $+1$, incorrect answers $-1$, and abstentions zero. This is an operational measure of the answer–abstain decision, rather than a claim that the model possesses calibrated self-knowledge. Our contributions are:

- **Reference-guided critique supervision.** Mirror-Critique turns solver trajectories and reference solutions into a supervised initialization for generative verification, using open-model critique synthesis and judgment-based rejection sampling.
- **Verifier reinforcement learning and aggregation.** We train the initialized verifier with balanced correctness labels and use repeated critique judgments to support weighted voting and selective abstention within one solver–verifier system.
- **Evaluation of accuracy and selective prediction.** Experiments at two model scales across five benchmarks compare answer accuracy, honesty scores, candidate-count scaling, and the quality of synthetic critiques. The SFT-to-RL comparisons show the contribution of verifier reinforcement learning beyond critique initialization alone.

## 2 Related Work

**Reinforcement learning for reasoning.** RLVR is central to reasoning-model training, including OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025). Studies of zero-RL examine how verifiable rewards develop reasoning behavior from base models (Li et al., 2025; Zeng et al., 2025), while other work applies RLVR to already distilled reasoning models (Luo et al., 2025). GRPO (Shao et al., 2024) estimates advantages from groups of sampled responses without a separate learned value model. It builds on PPO-style updates (Schulman et al., 2017) and has motivated variants such as DAPO, VAPO, and Dr. GRPO (Yu et al., 2025; Yue et al., 2025; Liu et al., 2025b). Adaptive sampling also changes which problems receive training effort (Yang et al., 2025). We use verifiable rewards for two roles: training a solver to produce answers and training a verifier to judge candidate answers. The latter requires a different response format and a different balance of positive and negative examples.

**Test-time sampling and aggregation.** Self-consistency selects the most frequent answer among sampled reasoning trajectories (Wang et al., 2023; Brown et al., 2024). Extensions adjust sampling or filter candidates (Aggarwal et al., 2023; Xue et al., 2023; Huang et al., 2024; Knappe et al., 2024). Learned reward models provide an alternative ranking signal (Yang et al., 2024b; Liu et al., 2024; 2025c), while Universal Self-Consistency prompts a model to compare responses (Chen et al., 2024). Set-level aggregators such as SSA and AggLM learn to produce an answer from a collection of solutions (Qi et al., 2025; Zhao et al., 2025). Mirror-Verifier instead scores candidates individually through generated critiques and then applies an explicit aggregation rule. This separates the learned verification behavior from the number of candidates available at test time.

**Critique and verification.** Process reward models learn to assess intermediate reasoning, including Math-Shepherd and Skywork-O1-PRM (Wang et al., 2024; He et al., 2024b). Critique fine-tuning supplies textual feedback rather than only scalar scores (Wang et al., 2025), and RISE combines generation and self-verification with outcome-verifier feedback (Liu et al., 2025a). Our approach uses reference-guided critiques as a cold start, followed by reinforcement learning on the correctness of the verifier's final judgment. The inference score is derived from repeated binary judgments, not a separately trained scalar reward head. Textual critiques remain inspectable, but

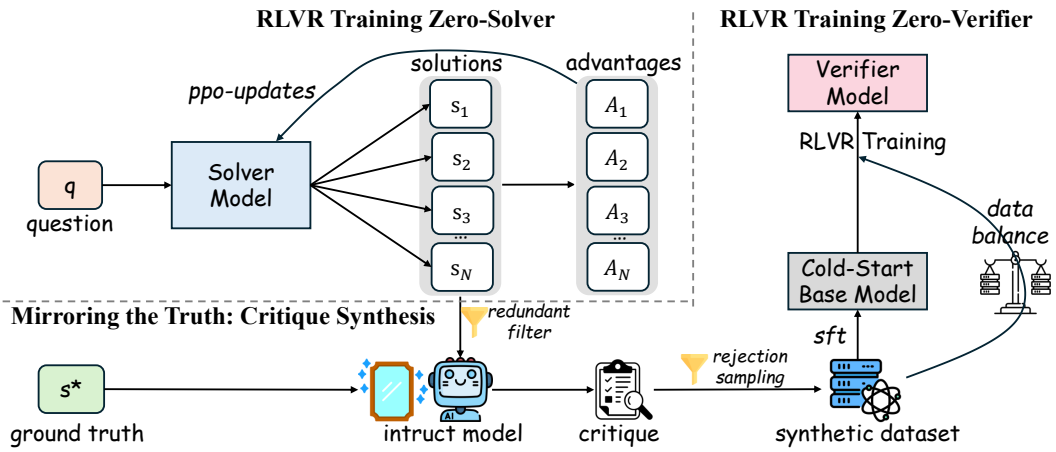



Figure 1: Mirror-Critique training pipeline. Solver RLVR supplies trajectories; an instruction-tuned model uses reference solutions to synthesize critiques. Judgment-based rejection sampling produces SFT data, and balanced-label RLVR refines the initialized verifier. Reference solutions are used during training-data construction, not verifier inference.

their correctness must be evaluated separately from the final judgment; a correct label does not certify every statement in the critique.

## 3 Mirror-Critique

### 3.1 Problem Formulation

Let $\mathcal{D}$ contain questions $q$, reference answers $a^*$, and reference solutions $s^*$ used for critique synthesis. A solver $S$ generates a solution $s$, including reasoning and an extracted final answer $a(s)$. A rule-based answer checker assigns the label $y(q, s) \in \{0, 1\}$ by comparing $a(s)$ with $a^*$. A verifier $V$ receives $(q, s)$ and generates a critique $c$ ending in a True/False judgment. Unlike a set-level aggregator, $V$ evaluates one candidate at a time.

At test time, the solver samples $K$ candidates $\{s_i\}_{i=1}^{K}$. A selection rule combines their answers and verifier scores to return $\hat{a}$, or an abstention symbol $\perp$. With abstention disabled, the objective is answer accuracy. With abstention enabled, we additionally measure the utility of choosing when to answer. The same verifier supports both modes; changing the abstention threshold does not require retraining the solver or verifier.

### 3.2 Mirroring the Reference for Critique Synthesis

We first train a Qwen2.5-Math base solver with RLVR and collect its generated trajectories. This reuses examples produced during solver training instead of requiring a separate solver-data generation stage. Math-Verify identifies answer-equivalent trajectories for deduplication. Because the equivalence check concerns final answers, this filtering reduces repeated outcomes rather than guaranteeing that every retained reasoning trace is unique.

For each retained candidate, Qwen2.5-7B-Instruct receives the question, candidate solution, and reference solution. It is asked to analyze the candidate step by step and conclude with a boxed True/False judgment. The reference makes this a contrastive task: the generator can compare an attempted derivation with a known solution instead of solving and verifying the problem from scratch. The synthesis prompt discourages explicit mentions of the reference in the resulting critique. The full prompt is provided in Appendix F.2.

We apply rejection sampling based on the final judgment. A critique is retained only when its extracted judgment agrees with $y(q, s)$, yielding a dataset $\mathcal{D}_{\text{synth}} = \{(q, s, c)\}$. This filter enforces consistency between the supervision label and the checked final answer. It does not prove that the

explanatory text is mathematically sound: a critique can reach the right True/False decision through an incorrect explanation. We therefore evaluate content quality separately in Section 4.6.

The supervision has two sources with distinct roles. Reference answers provide automatically checkable labels, while reference solutions help the instruction-tuned model articulate an explanation. The pipeline does not require a closed-model critique generator, but it does use an additional open instruction-tuned model and existing reference supervision. At deployment, neither the reference answer nor the reference solution is included in the verifier input.

### 3.3 Supervised Initialization and Verifier RLVR

**Cold start.** A base model may echo the prompt or continue the candidate solution instead of critiquing it. Such responses make the final-judgment reward a weak starting signal. We first minimize the negative log-likelihood of the synthetic critiques,

$$\mathcal{L}_{\mathrm{SFT}}(\phi) = -\mathbb{E}_{(q,s,c)\sim\mathcal{D}_{\mathrm{synth}}}\left[\log \pi_\phi(c \mid q, s)\right]. \tag{1}$$

The verifier learns to take a question–solution pair as input and generate an analysis followed by a judgment. We call this checkpoint **Mirror-SFT**. Importantly, the reference used by the synthesis model is not an input to the trained verifier. SFT transfers the critique behavior, rather than teaching the deployed verifier to depend on access to the answer key.

**Balanced correctness supervision.** Solver trajectories contain more incorrect than correct solutions. In an observed imbalanced setting, approximately 75% of examples are negative. Predicting False regardless of the input can therefore obtain substantial reward without meaningful verification. We sample positive and negative examples at a 1:1 ratio for verifier RLVR. This removes the reward advantage of the always-False strategy on the training distribution. The corresponding class-specific dynamics are shown in Appendix C.

To focus the verifier on settings where the correct answer is not dominant, we further select question–solution pairs from questions whose sampled solver accuracy is below 60%. This criterion uses the fraction of correct solver responses for the associated question, not the correctness of a single response. Balancing then preserves both correct and incorrect candidates for judgment learning. The filter targets difficult candidate distributions; it is not used to decide whether a test question should be answered.

**Reinforcement learning.** For a sampled critique $c$, the verifier receives a binary reward when its final judgment agrees with the known candidate label:

$$R_V(q, s, c) = \mathbf{1}\{\mathrm{judge}(c) = y(q, s)\}. \tag{2}$$

We optimize this reward with a GRPO-style update after SFT. The reward evaluates the final judgment, not the truth of every intermediate sentence or the pedagogical value of the explanation. Critique supervision and RLVR therefore play complementary roles: SFT supplies examples of the desired analysis format, and RLVR optimizes the correctness of the resulting decisions. The trained model is called **Mirror-Verifier**. Solver and verifier reward details appear in Appendix B.

### 3.4 Weighted Voting and Selective Abstention

For each candidate $s_i$, we independently sample $M$ critiques from the verifier and extract judgments $y_{i,j}$. The score is the fraction of positive judgments:

$$w_i = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\{y_{i,j} = \mathrm{True}\}. \tag{3}$$

This is a measure of agreement across sampled verifier outputs. It is not assumed to be a calibrated probability that the candidate is correct. Repeated sampling reduces dependence on any one generated critique, although shared model errors can remain across all samples.

Let $\mathcal{A}$ be the distinct final answers in the candidate set. We select

$$\hat{a} = \arg\max_{a\in\mathcal{A}} \sum_{i=1}^{K} (w_i + \beta)\mathbf{1}\{a(s_i) = a\}. \tag{4}$$

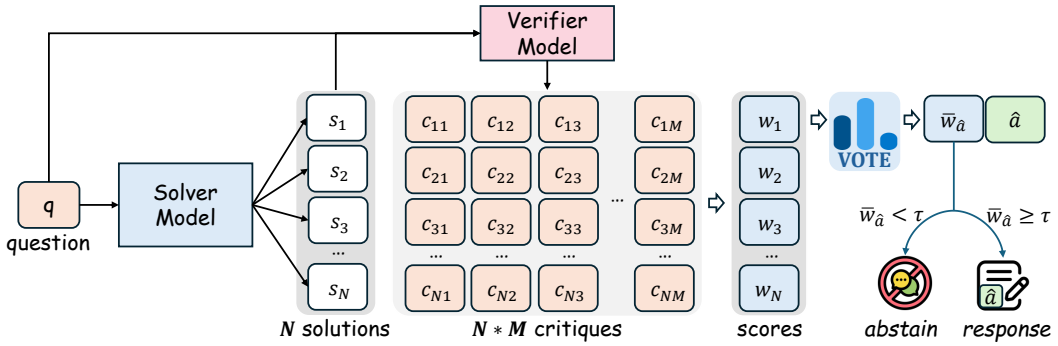



Figure 2: Inference with Mirror-Verifier. Each of $K$ candidate solutions (denoted $N$ in the diagram) receives $M$ critiques. Their judgments produce candidate scores for weighted voting. The selected answer is returned only if its mean verification score reaches $\tau$; $\tau = 0$ disables abstention.

The offset $\beta$ preserves an answer-frequency contribution even when verifier scores are small. Writing $n_a$ for the number of candidates with answer $a$ and $\bar{w}_a$ for their mean score, its total vote is $n_a(\bar{w}_a + \beta)$. This expression makes the role of verification explicit: a minority answer can overcome a larger group only if its score advantage compensates for its lower frequency. The method does not guarantee recovery of every minority-correct answer, and it cannot recover an answer absent from the candidate set.

Selection is followed by a separate abstention decision. For the winning answer,

$$\bar{w}_{\hat{a}} = \frac{\sum_{i=1}^{K} w_i \mathbf{1}\{a(s_i) = \hat{a}\}}{\sum_{i=1}^{K} \mathbf{1}\{a(s_i) = \hat{a}\}}, \qquad \text{output}(q) = \begin{cases} \hat{a}, & \bar{w}_{\hat{a}} \geq \tau, \\ \perp, & \bar{w}_{\hat{a}} < \tau. \end{cases} \tag{5}$$

Using the mean rather than the total vote separates confidence from the number of candidates supporting the answer. Otherwise, repeated low-scoring solutions could pass an abstention threshold simply because many were sampled. For $K = 1$, voting is trivial but the same verification-and-abstention rule still applies. The two sampling axes also have different purposes: $K$ increases the set of available solutions, whereas $M$ increases the number of critiques used to score each one.

# 4 Experiments

## 4.1 Experimental Setup

**Models and data.** We use Qwen2.5-Math at 1.5B and 7B scales (Yang et al., 2024a) for the solver and verifier. Training uses OpenR1-45K, a subset of OpenR1-Math-220k (Hugging Face, 2025). Qwen2.5-7B-Instruct synthesizes critiques for both scales; the resulting verifier training sets contain 170k and 116k instances, respectively. Evaluation covers AIME24, MATH-500 (Lightman et al., 2023), OlympiadBench (He et al., 2024a), AMC23, and MinervaMath (Lewkowycz et al., 2022). The aggregate results combine the five benchmarks, rather than taking an unweighted mean of the five displayed columns.

**Training and inference.** We change the RoPE base from 10,000 to 40,000 and extend the context window to 16,384 tokens. The RL objective omits the KL penalty and follows Dr. GRPO in removing response-length normalization and the standard-deviation normalization of advantages (Liu et al., 2025b). The reported learning rate is $10^{-7}$ throughout training. Solver and verifier batch sizes are 128 and 1,024, with 8 and 16 rollout samples, respectively. Training and test-time sampling use temperature 1.0. At inference, we generate $M = 16$ critiques per candidate and use $\beta = 0.15$ in weighted voting. The main tables use $K = 16$ solver candidates. Appendix A collects these settings.

**Baselines.** Pass@1 (Avg@16) averages correctness across 16 sampled responses per question; it is not the probability that at least one response is correct. Majority@$K$ selects the most frequent answer. Learned-verifier comparisons include Math-Shepherd-PRM (Wang et al., 2024), Skywork-O1-PRM (He et al., 2024b), and Qwen2.5-Math-7B-CFT (Wang et al., 2025), the latter trained on

Table 1: Answer accuracy (%) with 16 solver candidates and no abstention ($\tau = 0$). #Instances reports the training-data counts supplied for each verifier; it is not a training-compute measure. Overall denotes the reported aggregate over the combined benchmarks.

| Method / Verifier | #Instances | AIME24 | MATH-500 | Olympiad | AMC23 | Minerva | Overall |
|---|---|---|---|---|---|---|---|
| *Qwen2.5-Math-1.5B as the Solver* | | | | | | | |
| *pass@1 (avg@16)* | - | 11.9 | 75.0 | 39.6 | 44.2 | 31.1 | 49.2 |
| *majority@16* | - | 20.0 | 81.1 | 47.5 | 48.6 | 35.5 | 55.7 |
| Qwen2.5-Math-7B-CFT | 50k | 20.0 | 81.3 | 47.2 | 49.8 | 34.9 | 55.6 |
| Math-Shepherd-PRM | 445k | 20.0 | 81.4 | 47.6 | 48.2 | 35.7 | 55.8 |
| Skywork-o1-PRM-1.5B | unknown | 20.0 | 83.4 | 48.9 | **53.0** | 36.0 | 57.3 |
| Mirror-SFT-1.5B | 170k | 16.7 | 82.3 | 46.2 | 50.6 | 34.8 | 55.5 |
| **Mirror-Verifier-1.5B** | 170k | **23.3** | **84.0** | **49.5** | **53.0** | **37.9** | **58.2** |
| *Qwen2.5-Math-7B as the Solver* | | | | | | | |
| *pass@1 (avg@16)* | - | 23.2 | 84.2 | 46.7 | 57.1 | 38.1 | 57.3 |
| *majority@16* | - | 23.3 | 88.1 | 52.3 | 63.3 | 40.4 | 61.8 |
| Qwen2.5-Math-7B-CFT | 50k | 25.0 | 87.8 | 52.7 | 63.2 | 38.7 | 61.7 |
| Math-Shepherd-PRM | 445k | **26.7** | 88.9 | 52.4 | 62.7 | 40.0 | 62.0 |
| Skywork-o1-PRM-7B | unknown | **26.7** | 88.6 | 52.4 | **67.5** | 40.4 | 62.3 |
| Mirror-SFT-7B | 116k | 25.0 | 88.4 | 53.2 | 62.7 | 40.3 | 62.2 |
| **Mirror-Verifier-7B** | 116k | 25.0 | **89.1** | **54.1** | 63.9 | **41.2** | **63.0** |

50k GPT-4o-generated critique responses. Mirror-SFT is our supervised checkpoint before verifier RLVR and isolates the subsequent training stage within the same framework. Training-set sizes are reported for context, but do not establish compute-matched comparisons among these systems.

### 4.2 Evaluation Metrics

Let $C$, $W$, and $U$ denote the numbers of correctly answered, incorrectly answered, and unanswered questions in a dataset of size $n = C + W + U$. We report

$$\text{Accuracy} = \frac{C}{n}, \qquad \text{Honesty} = \frac{C - W}{n}. \tag{6}$$

Accuracy retains all questions in its denominator, including abstentions. Thus the threshold curves measure the fraction of all questions answered correctly, not accuracy conditional on answering. Honesty assigns utilities of $+1$, $-1$, and $0$ to a correct answer, an incorrect answer, and abstention, respectively. Tables and plots express these quantities on a scale multiplied by 100.

These definitions make the effect of abstention explicit. With no abstention, honesty equals $2\,\text{Accuracy} - 1$ and provides no additional ranking information. With abstention, let coverage be $(C + W)/n$; then honesty equals $2\,\text{Accuracy} - \text{Coverage}$. At fixed accuracy, a higher honesty score means fewer wrong answers are released. Rejecting every question gives zero utility, so a positive score requires more correct than incorrect answers, not merely frequent abstention. The metric operationalizes a particular equal-cost preference between correct and incorrect answers; it is not a general guarantee of calibration or safety.

### 4.3 Answer Accuracy without Abstention

Table 1 reports results with $\tau = 0$, so every question receives an answer. Mirror-Verifier obtains the highest reported aggregate accuracy at both model scales: 58.2% for the 1.5B solver and 63.0% for the 7B solver. The corresponding majority-voting results are 55.7% and 61.8%, giving improvements of 2.5 and 1.2 percentage points. These comparisons concern selection among sampled candidates, not improvements to the solver's underlying single-response policy.

For the 1.5B solver, Mirror-Verifier improves over majority voting on all five benchmarks. The gains are 3.3 points on AIME24, 2.9 on MATH-500, 2.0 on OlympiadBench, 4.4 on AMC23, and 2.4 on MinervaMath. It is best or tied for best in every benchmark column among the evaluated

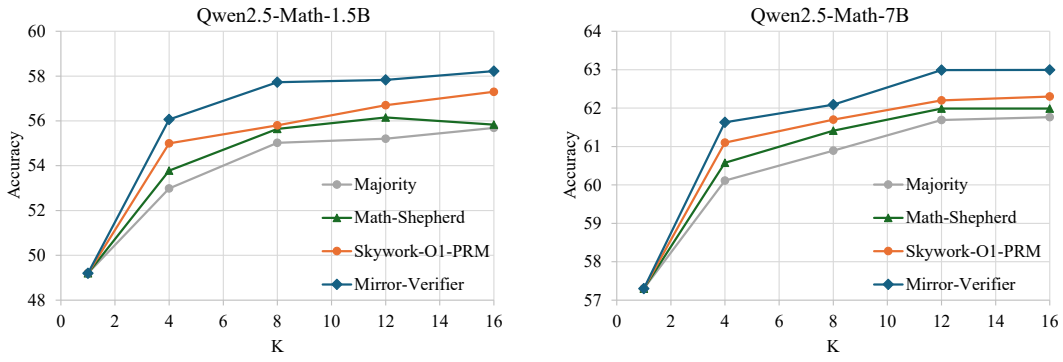



Figure 3: Answer accuracy as the number of solver candidates $K$ increases, combining the five benchmarks. Each panel compares aggregation methods for a fixed solver scale. Increasing $K$ expands the solution pool; verifier critique sampling remains a separate inference cost.

methods. Relative to the strongest aggregate learned-verifier baseline, Skywork-O1-PRM, its overall gain is 0.9 points. This is a smaller comparison than the gain over majority voting and should be distinguished from it.

For the 7B solver, the aggregate improvement is accompanied by gains on MATH-500, OlympiadBench, and MinervaMath, where Mirror-Verifier reaches 89.1%, 54.1%, and 41.2%. It does not win every benchmark: Math-Shepherd and Skywork reach 26.7% on AIME24 compared with 25.0%, and Skywork reaches 67.5% on AMC23 compared with 63.9%. The evidence therefore favors Mirror-Verifier in aggregate rather than establishing uniform superiority over every verifier on every task.

**Contribution of verifier RLVR.** Comparing Mirror-SFT with Mirror-Verifier holds the critique-based initialization pipeline fixed. For the 1.5B solver, aggregate accuracy rises from 55.5% to 58.2%; the SFT checkpoint alone is slightly below majority voting. For the 7B solver, the increase is from 62.2% to 63.0%. This comparison supports the role of the RL stage beyond learning to produce critique-shaped text. It does not separately identify how much of the improvement comes from balanced sampling, difficulty filtering, or the policy update, since those components are used together.

**Scaling the candidate set.** Figure 3 examines candidate counts up to 16. Mirror-Verifier remains competitive as the solution pool grows and yields the highest aggregate endpoint at both scales. The verifier is trained on individual question–solution pairs drawn from solver trajectories collected with eight rollouts per question. It can score 16 candidates at test time because the aggregation rule accepts a variable number of independently scored solutions. This is not a claim that the verifier was trained as an eight-element set model and extrapolates its context to a larger set.

The distinction matters when interpreting the curves. At $K = 1$, answer selection cannot change the solver's answer if abstention is disabled. At larger $K$, improvements can arise when score differences overturn an incorrect frequency-based choice. However, aggregate accuracy alone does not measure the recovery rate of minority-correct answers. The voting mechanism permits such recovery; a candidate-level analysis would be needed to quantify how often it occurs.

### 4.4 Selective Prediction across Benchmarks

Table 2 enables abstention with $\tau = 0.20$ for verifier-based methods. Pass@1 and majority voting are reported without abstention. Mirror-Verifier reaches aggregate honesty scores of 32.7 for the 1.5B solver and 31.3 for the 7B solver, exceeding the listed baselines. Relative to Mirror-SFT, the gains are 12.2 and 1.1 points, respectively. The larger 1.5B improvement suggests that the RL stage changes more than forced-choice answer ranking: its scores also support a better answer–abstain decision under this evaluation rule.

Table 2: Honesty scores ($100\times$ mean utility). Verifier-based methods use $\tau = 0.20$; Pass@1 and majority voting do not abstain. The Skywork variant is reproduced as reported for this evaluation.

| Method / Verifier | AIME24 | MATH-500 | Olympiad | AMC23 | Minerva | Honesty |
|---|---|---|---|---|---|---|
| *Qwen2.5-Math-1.5B as the Solver* | | | | | | |
| *pass@1 (avg@16)* | -76.2 | 50.0 | -20.8 | -11.6 | -37.8 | -1.60 |
| *majority@16* | -60.0 | 62.2 | -5.0 | -2.8 | -29.0 | 11.4 |
| Math-Shepherd-PRM | -60.0 | 62.6 | -4.89 | -3.61 | -27.9 | 11.7 |
| Skywork-o1-PRM-1.5B | -56.7 | 67.0 | 1.04 | 9.64 | -21.7 | 17.6 |
| Mirror-SFT-1.5B | -53.3 | 66.6 | 6.52 | 15.7 | -19.9 | 20.5 |
| **Mirror-Verifier-1.5B** | **-13.3** | **71.6** | **21.5** | **30.1** | **-5.15** | **32.7** |
| *Qwen2.5-Math-7B as the Solver* | | | | | | |
| *pass@1 (avg@16)* | -53.6 | 68.4 | -6.6 | 14.2 | -23.8 | 14.6 |
| *majority@16* | -53.4 | 76.2 | 4.60 | 26.6 | -19.2 | 23.6 |
| Math-Shepherd-PRM | -46.7 | 77.6 | 4.74 | 25.3 | -19.5 | 23.9 |
| Skywork-o1-PRM-1.5B | **-26.7** | 75.0 | 11.3 | **38.6** | -17.3 | 27.4 |
| Mirror-SFT-7B | **-26.7** | 77.0 | 16.7 | 30.1 | -16.9 | 30.2 |
| **Mirror-Verifier-7B** | **-26.7** | **78.0** | **17.4** | 33.7 | **-14.3** | **31.3** |

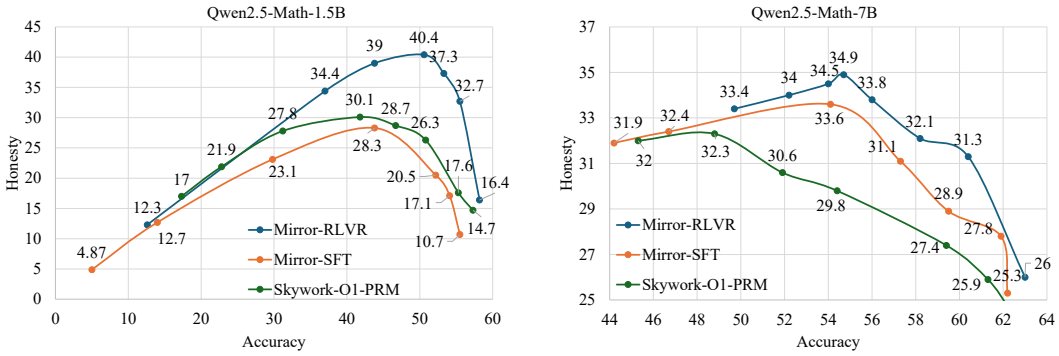



Figure 4: Honesty–accuracy curves obtained by varying the abstention threshold on the combined benchmarks. Accuracy counts correct answers over all questions, including rejected ones in the denominator. Mirror-RLVR denotes Mirror-Verifier. Curves compare operating points, not confidence calibration.

The per-benchmark results help separate these effects. For the 1.5B solver, AIME24 honesty improves from majority voting's $-60.0$ to $-13.3$, while AMC23 improves from $-2.8$ to 30.1. Mirror-Verifier still has negative utility on AIME24 and MinervaMath, meaning that the chosen threshold does not eliminate all unfavorable error patterns. At 7B, it obtains the best or tied-best score on four benchmarks, while Skywork has the higher AMC23 score. These patterns agree with the accuracy results in showing that aggregate gains need not imply dominance in every individual setting.

A common threshold is convenient for a single evaluation protocol, but scores from different verifier architectures need not have the same calibration. Accordingly, Table 2 establishes performance at the stated operating point, not at each method's individually optimized threshold. The comparison with non-abstaining majority voting also combines two changes: learned answer selection and the ability to reject an answer. The threshold curves below examine the trade-off more directly within the verifier-based systems.

### 4.5 The Accuracy–Abstention Trade-off

Figure 4 compares Mirror-Verifier, Mirror-SFT, and Skywork-O1-PRM as the threshold changes. Raising the threshold removes answers with low mean verification scores. This can improve honesty when rejected answers are mostly wrong, but it can also discard correct answers. As a result,

increasing abstention need not improve utility monotonically. The useful regime retains correct answers while reducing the number of incorrect ones.

The plotted Mirror-Verifier curve generally offers higher honesty at comparable accuracy than the SFT initialization. Under Equation 6, this means that fewer incorrect answers are returned for a similar number of correctly answered questions. The comparison is especially pronounced at 1.5B; the separation at 7B is smaller. This is consistent with the differences in Tables 1 and 2, where verifier RLVR gives a larger improvement at the smaller scale. The curves do not show that one fixed threshold will transfer to a new benchmark distribution without adjustment.

### 4.6 Quality of the Synthetic Critiques

Table 3: Critique-content quality on 30 randomly sampled examples from each synthetic dataset. Both datasets use Qwen2.5-7B-Instruct for critique generation; the suffix identifies the associated solver scale.

| Synthetic Data | DeepSeek-V3.1 | Human annotators |
|---|---|---|
| Mirror-Critique-1.5B | 83.3% | 80.0% |
| Mirror-Critique-7B | 80.0% | 76.7% |

Rejection sampling checks the critique's final judgment against the candidate answer label. To examine whether the accompanying explanation is also correct, we randomly sample 30 examples from each synthetic dataset and evaluate the critique content with DeepSeek-V3.1 and human annotation. Two annotators independently label each example, and a third resolves disagreements. Appendix D gives the protocol and Appendix F gives the model-judge prompt.

Table 3 reports human-judged correctness of 80.0% and 76.7% for the 1.5B- and 7B-associated datasets. These correspond to 24 and 23 correct critiques out of 30. The model judge marks 25 and 24 correct, respectively. Thus the audit supports the presence of useful explanatory supervision, while also exposing noise that survives the label filter. With only 30 examples per dataset, the one-example difference between the two judges is descriptive and should not be interpreted as a reliable measure of judge quality.

Appendix E illustrates a critique that rejects an incorrect answer and identifies the required numerical correction, but also contains an imprecise statement about a complement graph. This example reinforces the audit's distinction between final-judgment correctness and explanation quality: useful critique supervision is not equivalent to a collection of certified proofs.

**Inference cost and scope.** Mirror-Verifier trades additional generation for better aggregation. With $K = 16$ solver candidates and $M = 16$ critiques per candidate, a query requires 256 generated critiques in addition to the solver responses. Candidate count alone is therefore not an equal-compute comparison with majority voting or scalar reward models. Our experiments establish answer-quality and selective-prediction improvements at the reported sampling settings, rather than a latency or token-efficiency advantage. The method is most directly applicable when answer checking and reference solutions are available for constructing supervision.

## 5 Conclusion

Mirror-Critique trains a generative verifier by combining reference-guided critique synthesis, supervised initialization, and reinforcement learning on balanced correctness labels. The resulting Mirror-Verifier uses repeated critique judgments to score candidate solutions, with separate rules for selecting an answer and deciding whether to return it. Across five mathematical benchmarks and two model scales, it improves reported aggregate accuracy over majority voting and the evaluated learned-verifier baselines. Its scores also improve selective-prediction utility under an explicit reward for correct answers and penalty for incorrect ones. The results support critique-based training as a useful complement to outcome supervision, while the content audit shows why correct verifier judgments should not be equated with fully correct explanations.

## AI USE STATEMENT

Language models are part of the research method: open models are used for solver and verifier training and inference, Qwen2.5-7B-Instruct generates synthetic critiques, and DeepSeek-V3.1 assists with critique-quality evaluation. The manuscript also used Kimi-K2, DeepSeek-V3.2, and Gemini 3 for language polishing and grammar correction, and a generative AI assistant for organization and formatting during template conversion. The authors are responsible for checking AI-assisted text and for the scientific content and reported results.

## ETHICS STATEMENT

This work studies mathematical reasoning using existing datasets and includes human annotation of synthetic critique quality. The task does not require sensitive personal information. Generated critiques can contain errors, and the reported honesty score is an evaluation utility rather than a guarantee of truthfulness or safe deployment. Verifier outputs should not replace independent checking in consequential applications.

## REPRODUCIBILITY STATEMENT

Section 3 specifies the critique-generation pipeline, verifier reward, voting rule, and abstention criterion. Section 4 and Appendix A describe the models, datasets, training settings, and inference parameters. The appendix provides solver and verifier reward details, prompts, and the human critique-quality protocol. Benchmark-level tables accompany aggregate results to make the reported comparisons inspectable.

## REFERENCES

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

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 preprint arXiv:2407.21787*, 2024.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language models. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=LjsjHF7nAN.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu,

Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 open series, November 2024b. URL https://doi.org/10.5281/zenodo.16998085.

Siyuan Huang, Zhiyuan Ma, Jintao Du, Changhua Meng, Weiqiang Wang, and Zhouhan Lin. Mirror-consistency: Harnessing inconsistency in majority voting. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2408–2420, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.135. URL https://aclanthology.org/2024.findings-emnlp.135/.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Tim Knappe, Ryan Luo Li, Ayush Chauhan, Kaylee Chhua, Kevin Zhu, and Sean O'Brien. Enhancing language model reasoning via weighted reasoning in self-consistency. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=2w0CIzWlle.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards, 2025a. URL https://arxiv.org/abs/2505.13445.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acemath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint arXiv:2412.15084*, 2024.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. AceMath: Advancing frontier math reasoning with post-training and reward modeling. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3993–4015, Vienna, Austria, July 2025c. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.206. URL https://aclanthology.org/2025.findings-acl.206/.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

Jianing Qi, Xi Ye, Hao Tang, Zhigang Zhu, and Eunsol Choi. Learning to reason across parallel samples for llm reasoning. *arXiv preprint arXiv:2506.09014*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

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

Mingfeng Xue, Dayiheng Liu, Wenqiang Lei, Xingzhang Ren, Baosong Yang, Jun Xie, Yidan Zhang, Dezhong Peng, and Jiancheng Lv. Dynamic voting for efficient reasoning in large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3085–3104, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.203. URL https://aclanthology.org/2023.findings-emnlp.203/.

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, 2024a. URL https://arxiv.org/abs/2409.12122.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Dongchun Xie, Yiwei Wang, Xiaodan Liang, and Jing Tang. Depth-breadth synergy in rlvr: Unlocking llm reasoning gains with adaptive exploration, 2025. URL https://arxiv.org/abs/2508.13755.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan 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, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025. URL https://arxiv.org/abs/2504.05118.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Kulikov. The majority is not always right: Rl training for solution aggregation, 2025. URL https://arxiv.org/abs/2509.06870.

## A Training and Inference Configuration

Table 4 collects the numerical settings reported for the experiments. Solver rollouts and verifier rollouts belong to different training tasks: the former produce candidate solutions, while the latter produce critiques of a given candidate. At inference, $K$ counts solver solutions and $M$ counts critiques per solution. The configuration should therefore not be summarized by one undifferentiated sampling count.

Table 4: Reported configuration for Mirror-Critique.

| Setting | Value |
|---|---|
| Solver / verifier base family | Qwen2.5-Math |
| Model sizes | 1.5B, 7B |
| Training questions | OpenR1-45K |
| Critique synthesis model | Qwen2.5-7B-Instruct |
| Synthetic instances (1.5B / 7B) | 170k / 116k |
| RoPE base | 40,000 (changed from 10,000) |
| Context window | 16,384 tokens |
| Learning rate | $10^{-7}$ |
| Solver / verifier batch size | 128 / 1,024 |
| Solver / verifier training rollouts | 8 / 16 |
| Positive : negative verifier sampling | $1:1$ |
| Question-selection criterion for verifier RL | Sampled solver accuracy $< 60\%$ |
| Training / inference temperature | 1.0 / 1.0 |
| Test solver candidates in main tables, $K$ | 16 |
| Test critiques per candidate, $M$ | 16 |
| Weighted-vote offset, $\beta$ | 0.15 |
| Abstention threshold, accuracy table | 0 |
| Abstention threshold, honesty table | 0.20 |

## B Reward Functions

**Solver reward.** The solver receives a binary answer-correctness reward from Math-Verify. A response containing code is marked incorrect in this setup, since code execution and tool-integrated reasoning are outside the evaluated solver pipeline. The reward is attached to the final outcome and does not certify the intermediate derivation. A reference solution is used later for critique synthesis; it is not needed as a model-based reward for solver training.

**Verifier reward.** The verifier generates a textual critique ending in a boxed True/False judgment. A regular expression extracts the judgment, which is compared with the candidate's known correctness label to produce the binary reward in Equation 2. A response must provide the expected judgment to receive reward for agreement. This reward is different from the quality audit: the audit asks whether the critique content is correct, whereas RLVR rewards the terminal decision about the candidate's answer.

## C Class Imbalance and Training Dynamics

An imbalanced verifier dataset contains approximately 75% negative examples. Figure 5 shows the resulting class-specific behavior: an increasing tendency to predict False improves accuracy on negative candidates while harming accuracy on positive candidates. With balanced sampling, this one-sided pattern is reduced. Balancing removes a simple label-prior shortcut, but does not itself guarantee correct reasoning in the generated critique.

Figure 6 reports the reward and F1 trajectories for the 7B verifier. Both increase over the displayed training interval. Considering F1 alongside reward is useful here because aggregate correctness alone can conceal asymmetric behavior between positive and negative candidates. The held-out

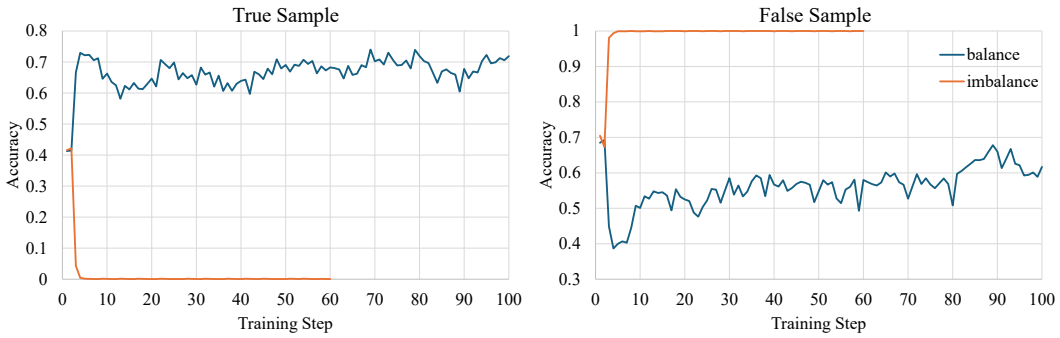



Figure 5: Verifier training with balanced and imbalanced labels. The panels report accuracy on False and True samples separately, making the collapse toward negative predictions visible.

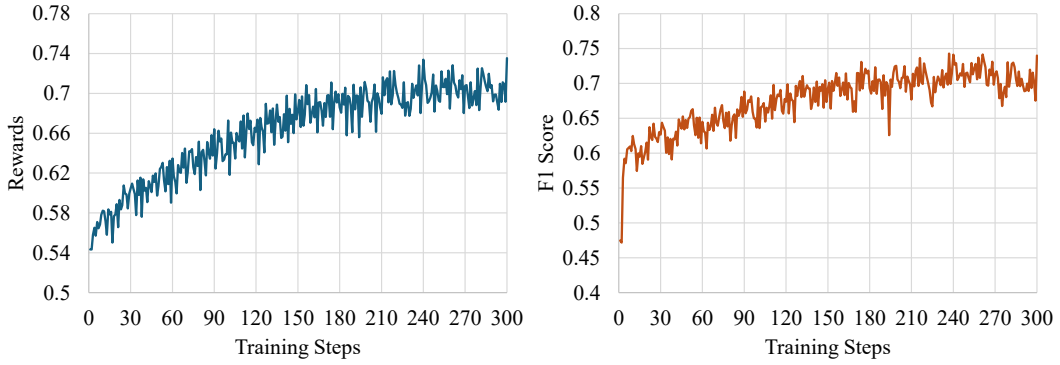



Figure 6: Mirror-Verifier-7B training dynamics: verification reward and training F1 score with balanced data. These are training diagnostics rather than held-out proof of generalization.

solver–verifier evaluations in the main text provide the complementary assessment of downstream answer selection.

## D Human Critique-Quality Annotation

We randomly sample 30 examples from each of the Mirror-Critique-1.5B and Mirror-Critique-7B synthetic datasets. Two students who scored within the top 2% in the college entrance examination independently annotate each example. A third student resolves disagreements. The annotators receive question–solution–critique triplets in Markdown format and label whether the critique content is correct. The percentages in Table 3 are the resulting binary correctness rates, not ratings of style or helpfulness. DeepSeek-V3.1 evaluates the same type of content using the prompt in Appendix F.

## E Critique Example and Cold-Start Behavior

Figure 7 reproduces a synthetic critique of a counting problem involving 16 teams. The candidate proposes 65 games, while the critique identifies the correction $120 - 64 = 56$ and returns False. The numerical reasoning can be understood by considering the complement of the played-game graph: it must be triangle-free, so at most 64 of the 120 possible pairs remain unplayed. However, the critique's first step refers to an independent set in the complement, which is an imprecise description of the condition. We retain the example to illustrate that useful corrections can coexist with flaws in the explanation.

**Question**

In a football championship, 16 teams participate, each playing against each other once. What is the minimum number of games that must be played so that among any three teams, there are two that have already played against each other?

**Candidate Solution**

1. **Understanding the Problem:**
  - We have 16 teams in a football championship.
  - Each team plays against each other team exactly once.
  - We need to find the minimum number of games such that among any three teams, there are two that have already played against each other.

2. **Graph Theory Representation:**
  - Represent the teams as vertices of a graph.
  - Represent a game between two teams as an edge between two vertices.
  - We need to ensure that in any subset of three vertices (teams), at least one edge (game) exists.
...

8. **Conclusion:**
  - Therefore, the minimum number of games that must be played so that among any three teams, there are two that have already played against each other is:
  \\[
  \\boxed{65}
  \\]

**Synthetic Critique**

Now I will give the step-by-step Analysis followed by the boxed judgment.
## Analysis:
1. **Understanding the Problem:**
  - The problem correctly identifies the need to ensure that among any three teams, there are two that have played against each other. This translates to a graph theory problem where we need to avoid an independent set of size 3 in the complement graph.

2. **Graph Theory Representation:**
  - The teams are represented as vertices, and the games as edges. This is correct.

3. **Combinatorial Approach:**
  - The problem correctly identifies the need to find the minimum number of edges such that the graph does not contain an independent set of size 3. This is correct.
...
8. **Minimum Number of Games:**
  - The solution incorrectly states that the minimum number of games required is 65. The correct calculation should be:
  \\[
  120 - 64 = 56
  \\]
  - The solution should have subtracted the maximum number of edges in a triangle-free graph from the total number of edges in a complete graph.

9. **Conclusion:**
  - The solution concludes with 65, which is incorrect. The correct answer is 56.

\\boxed{False}

Figure 7: Original synthetic critique example. The critique rejects the candidate's answer and gives the corrected count, but not every intermediate statement is precise. This illustrates why label agreement and critique-content quality are evaluated separately.

**Cold-start motivation.** Before critique SFT, we observe base-model responses that repeat the verification instructions or echo the candidate rather than analyze it. The following reproduced response illustrates instruction repetition. This motivates supervised initialization in the evaluated setup; it is not a claim that every base model is incapable of verification.

**Example of Instruction Repetition by the Base Model**

You are an expert mathematics tutor who always thinks step-by-step. You will be shown: Question and its Solution. Your task:
* Analyze the Solution according to the Question
* Produce a numbered step-by-step analysis of the Solution, explaining why it is correct or incorrect.
* End with a single line containing only
$\boxed{\text{True}}$ – if the boxed answer in the Solution is correct,
$\boxed{\text{False}}$ – otherwise.

# F PROMPTS

The prompts below distinguish reference-guided synthesis from inference. Only the synthesis model and the content-quality judge receive a reference. Solver and verifier deployment do not require access to a test answer key. The instruction to avoid mentioning the reference controls phrasing; it does not replace the separate requirement to exclude reference inputs at inference.

## F.1 SOLVER

**Prompt for Solving Reasoning Tasks**

Your task is to solve the given question step by step. You should conduct a systematic, thorough reasoning process before providing the final answer. This involves analyzing, summarizing, exploring, reassessing, and refining your reasoning process through multiple iterations. Each reasoning step should include detailed analysis, brainstorming, verification, and refinement of ideas. You should include the final answer in \boxed{} for closed-form results like multiple choices or mathematical results.

## F.2 REFERENCE-GUIDED SYNTHESIS

**Prompt for Critique with Ground Truth**

You are an expert mathematics tutor who always thinks step-by-step. You will be shown: Question, Ground Truth (hidden from the student), Solution. Your task:
* Analyze the Solution according to the Ground Truth. But do not mention 'ground truth', 'correct answer', 'official solution', etc.
* Produce a numbered step-by-step analysis of the Solution, explaining why it is correct or incorrect.
* End with a single line containing only
$\boxed{\text{True}}$ – if the boxed answer in the Solution is correct,
$\boxed{\text{False}}$ – otherwise.

## F.3 VERIFIER INFERENCE

**Prompt for Critique without Ground Truth**

You are an expert mathematics tutor who always thinks step-by-step. You will be shown: Question and its Solution. Your task:
* Analyze the Solution according to the Question
* Produce a numbered step-by-step analysis of the Solution, explaining why it is correct or incorrect.

* End with a single line containing only
$\boxed{\text{True}}$ – if the boxed answer in the Solution is correct,
$\boxed{\text{False}}$ – otherwise.

### F.4 CRITIQUE-CONTENT EVALUATION

**Prompt for Evaluating the Critique Content**

You are an evaluator tasked with analyzing critique accuracy. For each input, you will receive:
- Question: the problem statement
- Ground truth solution: the correct reference solution
- Candidate solution: a proposed solution to the question
- Critique: an analysis evaluating the candidate solution

Your task is to identify if the critique content is correct.

Process the following Input:
Question:
{question}

Ground truth solution:
{ground_truth_solution}

Candidate solution:
{candidate_solution}

Critique: {critique}

Now, please determine whether the critique to the candidate solution is accurate or not. Finally, provide your judgment in the specified boxed format. (Shorten your output and give me quick judgment)

## G SCOPE AND LIMITATIONS

Mirror-Critique is evaluated on mathematical problems with reference solutions and automatically checkable final answers. Extension to proof assessment or tool-integrated reasoning would require corresponding supervision and evaluation rules. The critique audit is small, and correct final judgments do not ensure faithful explanations. Repeated critique generation adds inference cost; the experiments do not establish an equal-compute advantage. Finally, the honesty metric uses a specified error penalty, so applications with different costs or data distributions may prefer a different abstention operating point.