SKILLCRITIC: SKILL-AWARE CALIBRATION OF GENERATIVE CRITICS FOR MATHEMATICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has substantially improved mathematical reasoning in large language models, but reliable credit assignment remains challenging when supervision is available only at the final answer. Generative critics provide denser intermediate feedback by evaluating partial solutions, yet we find that semantically flawed prefixes can still receive over-optimistic scores and that these errors often recur in recognizable forms. We therefore introduce **SKILLCRITIC**, a skill-aware calibration framework that organizes recurring reasoning failures into a compact bank of *failure skills*. SKILLCRITIC combines failure-skill activations with critic confidence to apply a bounded correction to actor-facing credit signals before policy optimization, while leaving the critic architecture and training target unchanged. Across seven mathematical reasoning benchmarks, SKILLCRITIC improves Avg7 from **35.17% to 37.14%** with the SFT Critic. With the RL-Post-Trained Critic, overall performance remains comparable to the matched generative-critic baseline, while AIME-family accuracy increases from **13.02% to 14.37%**. Controlled analyses show that the performance gap is substantially larger on skill-hit trajectories, while randomizing skill semantics weakens the AIME-family advantage. We further identify critic stability as a separate limitation during extended optimization. These results show that recurring failure skills provide a useful signal for improving the reliability of critic-derived policy credit.

1 INTRODUCTION

Reinforcement learning has become an effective approach for improving mathematical reasoning in large language models (LLMs). Methods based on verifiable outcome rewards, such as GRPO (Shao et al., 2024), have produced substantial reasoning gains, and large-scale RL post-training has further strengthened this paradigm (Guo et al., 2025). However, outcome-level supervision evaluates an entire solution only after generation is complete. For long reasoning trajectories, this provides limited information about which intermediate decisions should receive credit, making credit assignment a persistent challenge (Zhou et al., 2024; Feng et al., 2026). Recent work therefore introduces process reward models (Lightman et al., 2024; Wang et al., 2024; Luo et al., 2024) and generative critics such as GENAC (Shan et al., 2026) to provide denser feedback over intermediate reasoning steps.

Our Key Observation. We find that denser critic feedback can still be systematically over-optimistic. A partial solution may already contain a substantive reasoning error while receiving a high critic value because the local derivation remains plausible. More importantly, these errors exhibit recurring structure across problems. We repeatedly observe failure modes such as unjustified uniformity assumptions, invalid aggregation, risky algebraic simplifications, and missed depletion constraints. Figure 1 illustrates this phenomenon: an erroneous reasoning prefix can retain a high critic score even after a semantic failure has occurred.

These recurring errors provide a useful abstraction for credit calibration. We refer to them as *failure skills*. Each failure skill summarizes a family of semantically related reasoning errors that may appear under different surface forms. A compact skill bank can therefore associate diverse reasoning trajectories with a smaller set of recurring failure modes and provide a structured signal about when a critic estimate may be unreliable.

Why Existing Critic Signals Fall Short. Generative critics improve credit assignment by evaluating partial solutions at a finer granularity, but their values are still produced from the current reasoning

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.

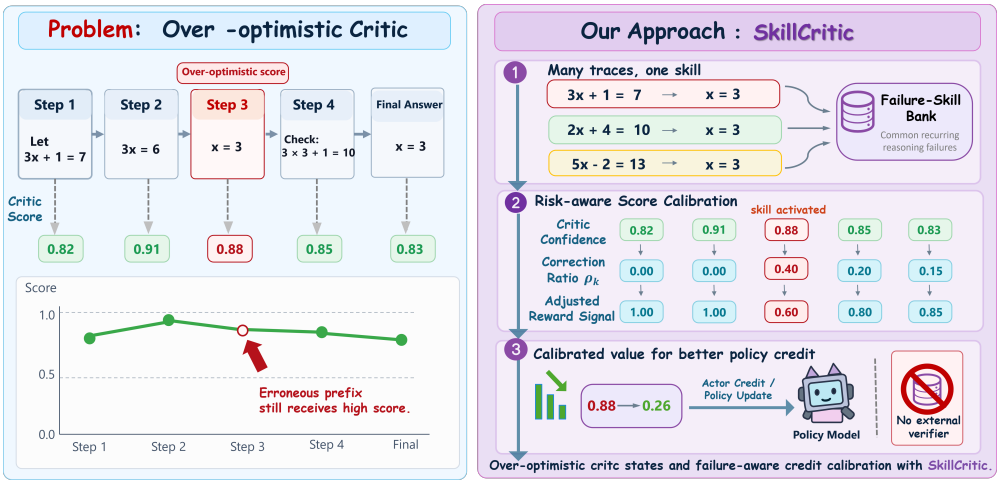

Figure 1: Over-optimistic critic states and failure-aware credit calibration with SKILLCRITIC.

context. Recent studies show that process-level evaluators can miss meaningful reasoning errors or respond to properties that only partially reflect semantic correctness (Ma et al., 2025; Zheng et al., 2025; Pandit et al., 2026). As a result, a locally coherent prefix can receive a confident value even when it matches a recurring failure mode. The resulting error is especially important during policy optimization because an overestimated segment can propagate overly positive credit to the corresponding tokens.

Research Question. *Can recurring reasoning failures be used to identify over-confident critic states and correct the resulting policy credit before optimization?*

To address this question, we propose **SKILLCRITIC**, a skill-aware calibration framework for generative critics. SKILLCRITIC maintains a compact bank of failure skills and combines their activation with critic confidence to apply a bounded correction to actor-facing credit before policy optimization, while leaving the raw critic prediction and training target unchanged. Across seven mathematical reasoning benchmarks, SKILLCRITIC provides broad improvements with the SFT Critic and more concentrated gains on challenging AIME-family problems with the RL-Post-Trained Critic. Controlled analyses further show that the performance gap is substantially larger on skill-hit trajectories, while extended critic optimization reveals a separate stability limitation.

Our contributions are summarized as follows:

- We identify a recurring failure mode in generative critics for mathematical reasoning: semantically flawed partial solutions can receive over-optimistic values, and these errors exhibit structured patterns across different problems.
- We introduce **SKILLCRITIC**, which organizes recurring reasoning errors into a compact bank of failure skills and combines their activation with critic confidence to selectively calibrate actor-facing credit before policy optimization.
- We evaluate SKILLCRITIC across seven mathematical reasoning benchmarks and controlled analyses. With the SFT Critic, SKILLCRITIC improves Avg7 from **35.17% to 37.14%**; with the RL-Post-Trained Critic, overall performance remains comparable while AIME-family accuracy, averaged over AIME24, AIME25, and AIME26, improves from **13.02% to 14.37%**. Further analyses localize the gain to failure-skill-relevant trajectories and identify critic stability as a separate limitation during extended optimization.

2 RELATED WORK

Reinforcement Learning and Credit Assignment for Reasoning. Reinforcement learning has become an effective approach for improving reasoning capabilities in large language models (Shao

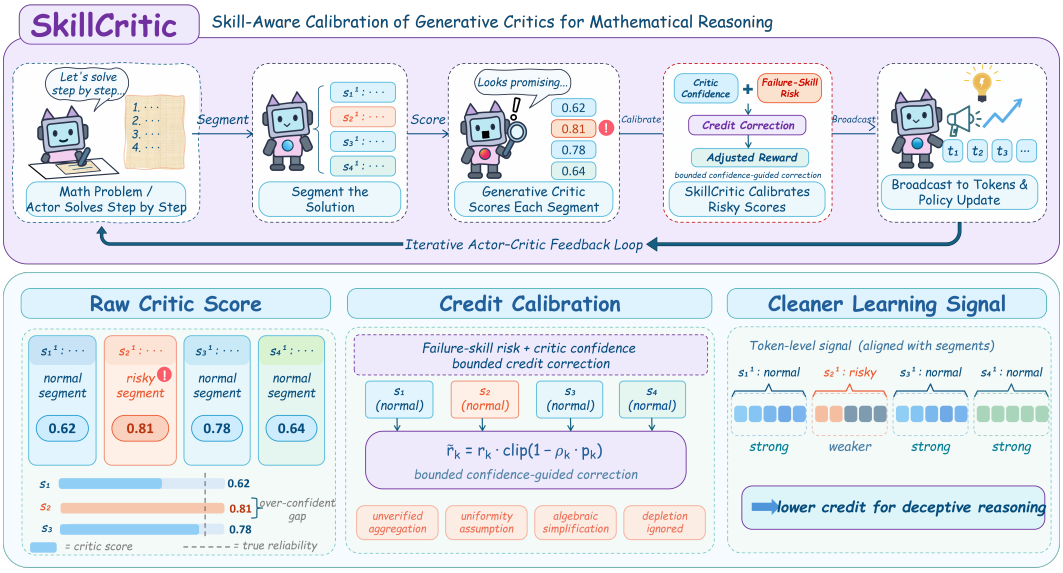

Figure 2: Overview of SKILLCRITIC. Failure skills and critic confidence jointly calibrate actor-facing credit before policy optimization.

et al., 2024; Guo et al., 2025). Recent studies further examine how model scale, training budget, and reward design affect RL post-training (Tan et al., 2026; Chen et al., 2026). A central challenge is credit assignment, since trajectory-level outcomes provide limited information about which intermediate decisions contribute to success or failure. Related work addresses this issue through hierarchical, turn-level, or finer-grained credit estimation in long-horizon settings (Zhou et al., 2024; Feng et al., 2026). Our work focuses on the reliability of critic-derived intermediate credit during reasoning-oriented reinforcement learning.

Value Modeling and Generative Critics. Actor–critic methods provide denser learning signals through learned value estimates, while recent generative critics improve value modeling by reasoning over partial solutions before predicting intermediate values (Shan et al., 2026). At the same time, analyses of step-level reward models suggest that intermediate scores may capture signals that only partially reflect semantic correctness or functional importance (Ma et al., 2025; Du et al., 2026). These observations motivate our focus on critic-derived credit calibration. SKILLCRITIC uses recurring failure skills as structured priors and combines their activation with critic confidence to selectively correct actor-facing credit when the critic may be over-optimistic.

Process Supervision and Structured Reasoning Signals. Process supervision provides finer-grained feedback for mathematical reasoning through human or automatically constructed step-level signals (Lightman et al., 2024; Wang et al., 2024; Luo et al., 2024; Sun et al., 2025). Recent benchmarks further show that reliable localization of reasoning errors remains challenging, especially on difficult mathematical problems (Zheng et al., 2025; Pandit et al., 2026). Skill-based RL methods also organize recurring behaviors into reusable abstractions (Xia et al., 2026). In SKILLCRITIC, the skill bank summarizes recurring reasoning failures and provides a structured risk signal for calibrating actor-facing credit within the existing generative-critic pipeline.

3 METHOD

We introduce SKILLCRITIC, a skill-aware calibration framework for generative critics. As illustrated in Figure 2, SKILLCRITIC identifies recurring reasoning failures and combines the resulting risk with critic confidence to calibrate actor-facing credit before policy optimization. We first present the motivating observation, followed by the problem setup, failure-skill construction, credit calibration, and integration with policy optimization.

3.1 Motivation: Recurring Failure Patterns in Critic Errors

Generative critics provide intermediate supervision by assigning values to partial reasoning trajectories. We inspect critic predictions on generated solutions and find that high critic scores do not always correspond to reliable reasoning states. A partial solution may already contain a semantic error while remaining locally coherent, causing the critic to continue assigning a high value.

Observation 1: Critic errors exhibit recurring structure. We inspect high-score trajectories according to their final correctness and find that recurring failure patterns are substantially more common when the final solution is incorrect. Section 5.1 provides a controlled analysis of this effect against randomized triggers.

Observation 2: Critic reliability is failure-dependent. Across different problems, the same critic can be reliable on many prefixes while repeatedly overestimating specific failure modes. These erroneous trajectories often share similar semantic structures despite substantial differences in their surface forms. Examples include unjustified uniformity assumptions, invalid aggregation, risky algebraic simplifications, and missed depletion constraints. These repeated errors suggest that critic reliability varies systematically with the type of reasoning failure.

Insight. Recurring failure modes provide an additional signal for judging when a critic value is likely to be unreliable. Their shared structure can be summarized into reusable abstractions that transfer across different problems and reasoning traces. We refer to these abstractions as *failure skills*, which provide structured risk signals when the same type of critic misjudgment reappears.

3.2 Preliminaries and Problem Setup

Generative Critic. Given an input problem x, an actor policy π_θ generates a reasoning response y, which is divided into K segments $[s_1, \ldots, s_K]$. At segment k, the generative critic evaluates the reasoning prefix $s_{\le k}$ and produces a scalar value

$$p_k = V_\psi(x, s_{\le k}), \qquad p_k \in [0, 1],$$

where a larger p_k indicates that the current prefix is judged more likely to lead to a successful solution. These segment-level values are subsequently converted into token-level learning signals by the underlying actor–critic optimizer.

Problem Setup. As observed in Section 3.1, some reasoning prefixes contain recurring semantic failures while still receiving high critic values. We use the failure-skill signal and critic prediction for different purposes: the former identifies structured reasoning risk, while the latter reflects critic confidence in the current prefix. Our goal is to combine these signals to correct actor-facing credit when a detected failure coincides with an over-confident critic state. The raw critic prediction and its training target remain unchanged.

3.3 Failure-Skill Bank

The observation in Section 3.1 suggests that critic errors can recur across problems even when the corresponding reasoning traces differ substantially in surface form. We capture this shared structure with a compact failure-skill bank

$$\mathcal{B} = \{b_1, \ldots, b_M\},$$

where each skill b_m represents a recurring semantic failure mode.

Our current bank contains four common patterns: unjustified uniformity assumptions, invalid aggregation, risky algebraic simplification, and missed depletion constraints. These patterns describe errors at the level of reasoning structure rather than specific tokens or problem instances. As a result, the same skill can be activated by different trajectories that exhibit a similar underlying failure.

For each skill b_m, we associate a detector D_m that examines the current prefix $s_{\le k}$ and produces a risk score

$$c_{k,m} = D_m(s_{\le k}), \qquad c_{k,m} \in [0, 1].$$

A larger $c_{k,m}$ indicates stronger evidence that the prefix exhibits failure skill b_m. Since multiple skills may apply to the same prefix, we summarize the strongest detected risk as

$$c_k = \max_{1 \le m \le M} c_{k,m}.$$

In our experiments, the detectors are implemented with lightweight textual and structural rules. Their role is to expose recurring failure structure to the calibration module; the following credit-calibration rule combines the resulting risk c_k with critic confidence.

3.4 SKILL-AWARE CREDIT CALIBRATION

Failure-skill activation determines where actor-side credit should be corrected, while critic confidence determines how strongly the correction should be applied. We combine these two signals through a bounded credit-calibration rule.

Given the raw critic score p_k, we first compute a confidence gate

$$g_k = \sigma\left(\frac{p_k - \tau_t}{0.05}\right),$$

where $\sigma(\cdot)$ denotes the sigmoid function and At each update, we compute the 80th percentile of critic scores in the current training batch and update τ_t by an exponential moving average.

To avoid aggressive correction early in training, we use a linear warm-up

$$w_t = \mathrm{clip}\left(\frac{t-15}{20}, 0, 1\right).$$

We further adapt the global correction scale according to recent activated samples. For each segment, we define

$$q_k = p_k c_k g_k,$$

and compute

$$\bar{q}_t = \frac{1}{|\mathcal{A}_t|}\sum_{k\in\mathcal{A}_t} q_k, \qquad \mathcal{A}_t = \{k : c_k > 0\}.$$

For batches containing activated segments, the running statistic is updated as

$$q_{\mathrm{ema},t} = 0.9 q_{\mathrm{ema},t-1} + 0.1\bar{q}_t.$$

The first valid batch initializes the running statistic directly, while batches without an activated segment leave it unchanged. The adaptive scale is then

$$\lambda_t = \frac{0.08}{q_{\mathrm{ema},t} + 10^{-4}}.$$

The final correction ratio is

$$\rho_k = \mathrm{clip}\left(\lambda_t w_t c_k g_k,\, 0,\, 0.5\right).$$

Rather than modifying the critic prediction, SKILLCRITIC applies this correction to the actor-facing reward signal:

$$\widetilde{r}_k = r_k(1 - \rho_k).$$

When no failure skill is activated, $c_k = 0$ and the original reward is preserved. When a detected failure coincides with high critic confidence, the corresponding actor-facing signal is reduced, with the correction bounded at 50%. The raw critic prediction p_k and the critic training target remain unchanged.

3.5 INTEGRATION WITH POLICY OPTIMIZATION

Actor Update. We use r_k to denote the segment-level actor-facing reward produced by the matched generative-critic pipeline before return and advantage construction. SKILLCRITIC does not redefine the baseline reward construction; it only replaces r_k with $\widetilde{r}_k = r_k(1 - \rho_k)$ on detected segments. The resulting reward sequence is then processed by the same return, GAE, and PPO optimization procedure as the matched baseline, while the terminal task reward and critic training target remain unchanged.

SKILLCRITIC operates on the actor-facing reward path before downstream credit estimation. For a detected segment s_k, the original reward r_k is replaced by its corrected counterpart $\widetilde{r}_k = r_k(1 -$

ρ_k).The resulting segment-level signal is broadcast to the corresponding tokens and used by the same GAE (Schulman et al., 2015) and PPO-style optimization procedure (Schulman et al., 2017) as the underlying generative-critic framework. The actor objective and optimization rule remain unchanged; SKILLCRITIC modifies only the credit signal supplied to actor optimization.

Critic Update. SKILLCRITIC does not alter the critic architecture, objective, or training target. In both experimental settings, the critic continues to be updated online using the same REINFORCE-style objective and original training signal as the matched GENAC-STYLE baseline. The two settings differ only in critic initialization: the SFT Critic starts from the supervised fine-tuning checkpoint, while the RL-Post-Trained Critic starts from a checkpoint obtained after critic RL post-training. Skill-aware calibration affects only the actor-facing reward path and is not used as the critic training target. Further training details are provided in the Appendix.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation Details. We use Qwen3-8B-Base (Yang et al., 2025) as the backbone for all experiments. RL training is conducted on a DeepSeekMath-style reasoning corpus with 40,306 prompts after preprocessing. All methods use the same actor initialization, prompt format, exact-answer reward, and evaluation protocol. For generative-critic methods, we consider two critic initializations. The *SFT Critic* starts from a critic checkpoint obtained after supervised fine-tuning, while the *RL-Post-Trained Critic* starts from a checkpoint selected from the later stage of a 120-step critic RL run. In both settings, the critic continues to be updated during actor training under the same critic objective; the distinction lies only in its initialization. Further implementation and optimization details are provided in the Appendix.

Baselines. We compare SKILLCRITIC with GRPO (Shao et al., 2024), PPO (Schulman et al., 2017), and a matched GENAC-STYLE baseline. GRPO serves as a critic-free reference, while PPO provides a conventional actor–critic baseline. We implement GENAC-STYLE with the same segment-level critic scoring and actor-side credit construction used in our framework. Since no official GENAC implementation is available, we treat it as a matched reimplementation. RandomSkill-B replaces the semantic regex contents of the failure-skill detectors while keeping the remaining calibration procedure unchanged. The hard-veto variant instead applies a fixed subtractive correction, $\mathrm{clip}(v - 0.3, 0, 1)$, without the confidence gate. Both are used only in the analysis experiments.

Evaluation Benchmarks and Metrics. We evaluate on AIME24, AMC23, AIME25, AIME26, MATH-500 (Lightman et al., 2024), OlympiadBench (He et al., 2024), and Minerva Math (Lewkowycz et al., 2022). For each problem, we sample 16 responses with temperature 0.6 and top-p 0.95 and report pass@1 under the avg@16 evaluation protocol. *Avg7* denotes the unweighted mean across all seven benchmarks, while *AIME-family* denotes the mean over AIME24, AIME25, and AIME26. The SFT Critic comparison uses an 80-step training window, whereas the RL-Post-Trained Critic comparison uses a 120-step training window. For each method, we report the best evaluation checkpoint within the corresponding window. Additional evaluation and optimization details are provided in the Appendix.

4.2 MAIN RESULTS

Table 1 reports the best-checkpoint results across the two critic settings.

SFT Critic. With the SFT Critic, SKILLCRITIC reaches 37.14% Avg7, compared with 35.17% for the matched GENAC-STYLE baseline and 36.24% for PPO. It also improves AIME-family accuracy from 10.72% to 13.47%, including an increase from 12.04% to 16.58% on AIME24.

RL-Post-Trained Critic. With the RL-Post-Trained Critic, SKILLCRITIC and GENAC-STYLE obtain similar Avg7 scores of 37.71% and 37.77%. The difference is concentrated on AIME-family problems: SKILLCRITIC reaches 20.12% on AIME24 versus 16.00% for GENAC-STYLE, increasing the AIME-family average from 13.02% to 14.37%. Section 5 examines where these differences arise.

Table 1: Main results with the SFT Critic and RL-Post-Trained Critic. All entries are pass@1 (%).

Method	AIME24	AMC23	MATH-500	Olympiad	Minerva	AIME25	AIME26	AIME-Fam.	Avg7
SFT Critic									
Base	9.79	47.19	68.86	35.36	36.08	8.13	7.92	8.61	30.48
GRPO	12.92	52.34	73.50	38.75	37.96	11.87	8.13	10.97	33.64
PPO	12.92	**57.81**	77.22	41.99	40.81	14.17	8.75	11.95	36.24
GENAC-STYLE	12.04	54.28	76.30	41.91	41.51	11.83	8.29	10.72	35.17
SKILLCRITIC	**16.58**	57.56	**78.18**	**42.22**	**41.61**	**14.21**	**9.63**	**13.47**	**37.14**
RL-Post-Trained Critic									
Base	9.79	47.19	68.86	35.36	36.08	8.13	7.92	8.61	30.48
GRPO	12.71	55.94	74.19	39.19	39.38	11.46	8.33	10.83	34.46
PPO	13.13	56.72	78.55	42.87	41.29	13.33	**9.79**	12.08	36.53
GENAC-STYLE	16.00	**60.06**	**79.80**	**44.01**	41.49	**14.13**	8.92	13.02	**37.77**
SKILLCRITIC	**20.12**	57.72	78.15	43.31	**41.69**	13.57	9.42	**14.37**	37.71

AIME-Fam. denotes the mean over AIME24, AIME25, and AIME26, and *Avg7* denotes the unweighted mean across all seven benchmarks. Each benchmark is evaluated with 16 samples per problem under the avg@16 protocol.

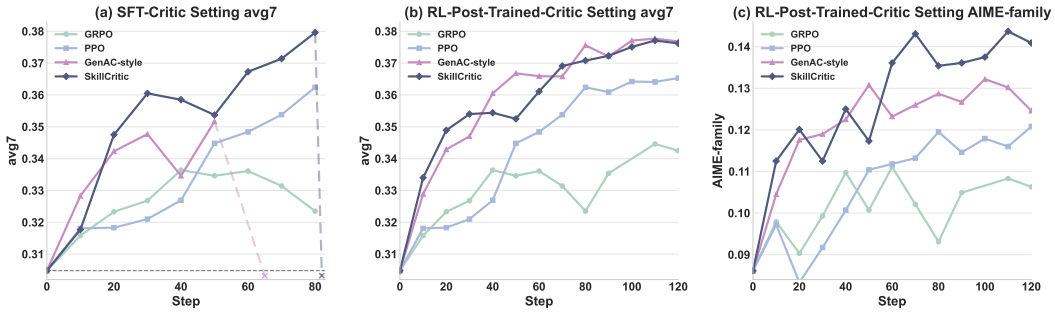

Figure 3: Training performance with the SFT Critic and RL-Post-Trained Critic.

5 ANALYSIS

We analyze whether critic errors exhibit recurring semantic structure, whether the gains depend on failure-skill relevance, and how critic stability evolves during extended optimization.

5.1 ARE CRITIC ERRORS STRUCTURED?

We partition trajectories by critic score and final correctness and compare failure-skill activation against randomly shuffled triggers, which estimate the background activation rate.

Key Observation. As summarized in Table 2 and visualized in Figure 4, the strongest concentration appears in high-score incorrect trajectories. The failure-skill bank is activated on 61.30% of these trajectories, compared with 19.40% under randomized triggers, yielding a 3.16× enrichment. Low-score incorrect trajectories also show elevated coverage at 34.60%, with a 1.75× enrichment. The corresponding enrichment is substantially weaker on correct trajectories, reaching 1.25× for the high-score group and 1.07× for the low-score group.

Implication. This pattern shows that failure-skill activation is most selective when the critic assigns a high score to a trajectory that ultimately fails. This is precisely the region in which over-optimistic critic values can introduce misleading policy credit. The recurring failure structure therefore provides an additional signal for identifying critic estimates whose reliability is not captured by the scalar value alone.

Table 2: Failure-skill coverage.

Trajectory subset	Skill bank	Random	Ratio
High-score & wrong	61.30%	19.40%	3.16×
High-score & right	22.70%	18.10%	1.25×
Low-score & wrong	34.60%	19.80%	1.75×
Low-score & right	11.60%	10.80%	1.07×

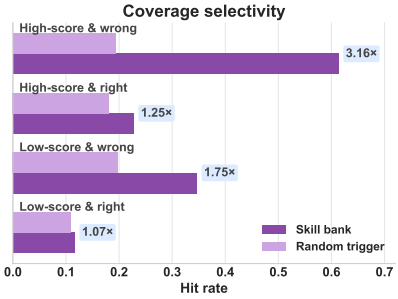

Figure 4: Failure-skill selectivity.

5.2 Do the Gains Depend on Failure-Skill Relevance?

The previous analysis shows that failure-skill activations are concentrated in critic errors. We next examine whether the downstream gain depends on the semantic relevance of these skills. We consider two complementary analyses: replacing the semantic contents of the failure-skill detectors and separating trajectories according to whether a failure skill is activated.

Randomized-Skill Control. RandomSkill-B replaces the semantic regex contents of the failure-skill detectors while keeping the detector structure, confidence gate, adaptive correction scale, and downstream calibration procedure unchanged. As shown in Table 3(a), the largest difference appears on AIME24, where SkillCritic reaches 20.12%, compared with 16.38% for RandomSkill-B and 16.00% for GenAC-Style. This difference raises the AIME-family average to 14.37%, compared with 13.14% and 13.02%, respectively. The effect varies across the three AIME benchmarks: RandomSkill-B is slightly higher on AIME25, while SkillCritic retains a smaller advantage on AIME26. Avg7 remains close across all three methods.

Hit-Conditioned Analysis. We next separate AIME-family trajectories according to whether at least one failure skill is activated. Table 3(b) shows a much larger performance gap on skill-hit trajectories: SkillCritic reaches 23.60% pass@1, compared with 18.10% for GenAC-Style. On non-hit trajectories, the corresponding scores are 11.90% and 11.60%. The gap therefore increases from 0.30 percentage points on non-hit trajectories to 5.50 points when a failure skill is activated.

Implication. The two analyses show that the effect of SkillCritic is concentrated in skill-relevant reasoning cases. Randomizing skill semantics weakens the advantage on AIME24 and the AIME-family aggregate, while the performance gap becomes substantially larger on trajectories where a failure skill is activated. This pattern is consistent with a targeted calibration effect rather than a uniform improvement across benchmarks or trajectories, although the hit-conditioned subsets may differ in trajectory composition across policies.

Table 3: Failure-skill relevance analysis.

(a) Randomized-skill control

Method	AIME24	AIME25	AIME26	AIME-Fam.	Avg7
SkillCritic	**20.12**	13.57	**9.42**	**14.37**	37.71
RandomSkill-B	16.38	14.08	8.96	13.14	37.61
GenAC-Style	16.00	**14.13**	8.92	13.02	**37.77**

(b) Hit-conditioned analysis

Subset	SkillCritic	GenAC-Style	Gap
Skill-hit	23.60	18.10	5.50
Non-hit	11.90	11.60	0.30

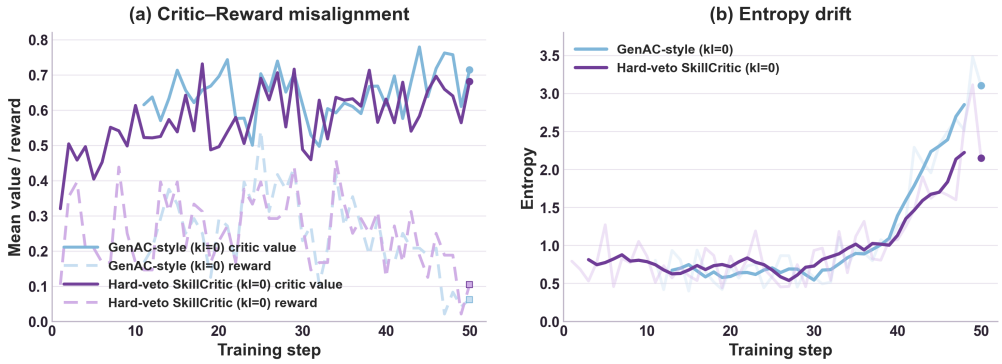

Figure 5: Critic–reward misalignment and entropy drift during extended optimization.

5.3 How Stable Are Critic Signals under Extended Training?

The narrower aggregate separation with the RL-Post-Trained Critic motivates examining how critic reliability evolves during extended optimization.

Critic Reliability Drifts. As shown in Figure 5(a), critic estimates increasingly separate from realized rewards as training approaches the unstable stage. For the matched GenAC-Style run, the mean critic value reaches 0.714 while the realized reward falls to 0.062. The hard-veto variant exhibits a similar discrepancy, with a mean critic value of 0.681 and a realized reward of 0.104. Although the critic outputs remain bounded in $[0, 1]$, their correspondence with observed outcomes deteriorates substantially. Strongly negative explained variance in the failed runs further supports this loss of predictive reliability rather than numerical divergence of the critic scores.

Training Dynamics Become Unstable. This degradation is accompanied by broader changes in training dynamics. Figure 5(b) shows that entropy remains around 0.8 during the earlier stage of training but rises sharply near instability, reaching 3.48 for GenAC-Style and 3.11 for the hard-veto variant. In the hard-veto run, the valid critic-output rate also decreases from 97.00% to 55.80%, while the average response length falls from approximately 630 to 421 tokens.

Implication.

These diagnostics identify critic stability as a separate bottleneck in extended generative-critic optimization. SkillCritic corrects actor-side credit associated with recurring over-optimistic critic states but does not stabilize the critic itself. The current evidence does not establish instability as the sole cause of the narrower RL-Post-Trained Critic separation, but it shows that credit calibration and critic reliability are complementary problems. Additional stabilization results are reported in the Appendix.

6 Conclusion

This work studies the reliability of generative critics for mathematical reasoning and identifies a recurring failure mode in which semantically flawed partial solutions receive over-optimistic intermediate values. We show that these errors exhibit reusable structure across trajectories and introduce SkillCritic, which organizes recurring reasoning failures into failure skills and combines their activation with critic confidence to calibrate actor-facing credit before policy optimization. Across seven mathematical reasoning benchmarks, SkillCritic improves Avg7 from 35.17% to 37.14% with the SFT Critic. With the RL-Post-Trained Critic, overall performance remains comparable to the matched baseline, while AIME-family accuracy increases from 13.02% to 14.37%. Further analysis shows that failure-skill activations are concentrated in high-score incorrect trajectories and that the downstream performance gap is substantially larger on skill-hit trajectories. We also identify critic stability as a separate issue during extended optimization, where predicted values can become increasingly misaligned with realized rewards. Overall, these results show that recurring failure skills provide a useful signal for improving the reliability of critic-derived policy credit, while the quality of the underlying generative critic remains an important factor in sustained RL training.

AI USE STATEMENT

During manuscript preparation, we used an LLM-based assistant for language editing, LaTeX troubleshooting, and local organization of manuscript and supplementary materials. All technical content, experimental results, and final manuscript decisions were reviewed by the authors for accuracy and consistency.

ETHICS STATEMENT

SKILLCRITIC is designed to improve the reliability of reinforcement learning for mathematical reasoning by correcting actor-side credit associated with recurring reasoning failures. Our experiments are conducted on mathematical reasoning benchmarks using model-generated trajectories and do not involve human-subject data or deployment in safety-critical environments. The proposed failure-skill detectors are used only as training-time signals for reasoning analysis and credit calibration. All automated assistance used in preparing the manuscript was reviewed by the authors for factual accuracy and compliance with venue policies.

REPRODUCIBILITY STATEMENT

SKILLCRITIC is implemented as an actor-side credit-calibration module within an existing generative-critic reinforcement learning pipeline. It does not modify the critic architecture, critic training target, or underlying actor optimization objective. Across matched comparisons, methods share the same actor initialization, task data, prompt format, reward function, rollout protocol, evaluation procedure, and critic training objective within each experimental setting. The complete method specification, including failure-skill detection, confidence gating, adaptive correction scaling, and actor-side reward calibration, is described in Section 3 and the Appendix. The source code, training configurations, and evaluation scripts will be released with the camera-ready version.

LIMITATIONS

Failure-Skill Coverage. Our current failure-skill bank contains a small set of manually specified reasoning patterns detected with lightweight textual and structural rules. These patterns cover the recurring failure modes examined in this work, but do not capture reasoning errors outside the predefined bank. Extending the bank automatically or replacing rule-based detection with learned failure-skill detectors remains a natural direction for future work.

Model and Benchmark Scope. Our experiments use Qwen3-8B-Base and focus on mathematical reasoning across seven benchmarks. While these benchmarks cover a range of competition and general mathematical reasoning tasks, evaluations at other model scales and in domains such as coding or theorem proving remain outside the scope of the current study.

Critic Stability. SKILLCRITIC corrects actor-side credit associated with detected failure skills but does not directly stabilize the underlying generative critic. Our analysis shows that critic reliability can degrade during extended optimization even when the correction mechanism remains bounded. Improving critic stability is therefore complementary to failure-skill-based credit calibration.

REFERENCES

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? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Kevin Du, Alexander Hoyle, Laura Ruis, and Acyr Locatelli. Legibility is not interpretability: Comparing judged and actual importance in chain-of-thought reasoning. *arXiv preprint arXiv:2609.04194*, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, 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, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Yiran Ma, Zui Chen, Tianqiao Liu, Mi Tian, Zhuo Liu, Zitao Liu, and Weiqi Luo. What are step-level reward models rewarding? counterintuitive findings from mcts-boosted mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24812–24820, 2025.

Shrey Pandit, Austin Xu, Xuan-Phi Nguyen, Yifei Ming, Caiming Xiong, and Shafiq Joty. Hard2verify: A step-level verification benchmark for open-ended frontier math. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22502–22517, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Zikang Shan, Han Zhong, Liwei Wang, and Li Zhao. Bringing value models back: Generative critics for value modeling in llm reinforcement learning. *arXiv preprint arXiv:2604.10701*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wei Sun, Qianlong Du, Fuwei Cui, and Jiajun Zhang. An efficient and precise training data construction framework for process-supervised reward model in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4292–4305, 2025.

Zelin Tan, Hejia Geng, Xiaohang Yu, Mulei Zhang, Guancheng Wan, Yifan Zhou, Qiang He, Xiangyuan Xue, Heng Zhou, Yutao Fan, et al. Scaling behaviors of llm reinforcement learning post-training: An empirical study in mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31300–31319, 2026.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024. *URL https://arxiv. org/abs/2312.08935*, 2(3), 2024.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

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

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1024, 2025.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024.

A Implementation and Evaluation Details

A.1 Data and Reward

All methods use the same DeepSeekMath-style mathematical reasoning corpus with the corrected chat template used in the main paper. The archived metadata records 40,315 source prompts before filtering and 40,306 prompts after one `filter_overlong_prompts` pass under the 1024-token prompt cap. The training set and preprocessing rule are therefore shared across GRPO, PPO, GenAC-Style, and SkillCritic.

All comparisons use the same exact-answer reward implemented with `math_verify`. No run in the main comparison introduces an external reward model.

A.2 Shared Training Configuration

Table 4 summarizes the training configuration shared across the main RL experiments.

Table 4: Shared training configuration for the main RL experiments.

Component	Setting
Base model	Qwen3-8B-Base
Training / validation	`deepscaler_train_chatfix.parquet` / `math500_chatfix.parquet`
Length caps	1024 prompt tokens; 4096 response tokens
Actor / critic optimizer	AdamW; learning rate 1×10^{-6}
PPO schedule	1 PPO epoch; clip ratio 0.2; grad clip 1.0
Rollout	$n = 6$; temperature 1.0; top-p 1.0; top-$k = -1$
Actor entropy coefficient	0.001
Critic segmentation	at most 128 tokens per segment; up to 768 new critic tokens
Critic decoding	greedy scoring; temperature 0; strict score parsing
Checkpoints	every 10 steps
Trainer / hardware	verl 0.2.0 + Ray 2.43; 8×A100 (4 actor, 4 critic)

A.3 Method Differences

All methods use the shared settings above and a KL coefficient of 0.001. Table 5 lists only the settings that differ across methods; RLPT denotes the Stage-2 RL-pretrained critic initialization.

Table 5: Method-specific differences in the main comparison.

Method	Critic initialization	Batch / rollout n	Skill correction
GRPO	none	8 / 6	none
PPO	value head from base	32 / 2	none
GenAC-noRLPT	SFT critic	8 / 6	none
SkillCritic with SFT Critic	SFT critic	8 / 6	soft multiplicative
GenAC-Style-RLPT	RLPT critic	8 / 6	none
SkillCritic-RLPT	RLPT critic	8 / 6	soft multiplicative
RandomSkill-B	RLPT critic	8 / 6	randomized triggers

The RL-Post-Trained-Critic setting differs from the SFT-Critic setting only in critic initialization. The RLPT runs start from a Stage-2 RL-pretrained critic, whereas the SFT setting starts from the SFT critic. The archived runs identify the SFT critic as the configuration that exhibits the early step-65 collapse discussed in the main text.

A.4 Evaluation Protocol

All core claims use the finalized avg@16 protocol. Each benchmark is evaluated with 16 samples per problem, and each checkpoint is evaluated in one seven-benchmark pass. The benchmark suite is AIME24, AMC23, Math500, OlympiadBench, Minerva, AIME25, and AIME26. We report

$$\text{AIME-Fam.} = \frac{\text{AIME24} + \text{AIME25} + \text{AIME26}}{3}, \qquad \text{Avg7} = \frac{1}{7}\sum_{b=1}^{7} \text{pass@1}_b.$$

For the main-text comparison, the SFT-Critic discussion is capped at step 80 and the RL-Post-Trained-Critic discussion is capped at step 120.

B SkillCritic Details

B.1 Segment Scoring

Given a problem x and a generated response y, the actor response is partitioned into contiguous segments

$$y = [s_1, s_2, \ldots, s_K],$$

with a maximum width of 128 tokens per segment. For each prefix $s_{\le k}$, the generative critic evaluates the partial solution and outputs a scalar score $p_k \in [0, 1]$ that estimates the chance that the current prefix will eventually lead to a correct solution.

The critic emits a short rationale followed by a strict terminal score block,

$$\texttt{<final><score>}N\texttt{</score></final>}, \qquad N \in \{0, 1, \ldots, 10\}.$$

The parsed value is normalized to $[0, 1]$. All main experiments use the same fail-closed parser.

B.2 Failure Skill Bank

The current failure-skill bank contains four recurring reasoning failures. Table 6 lists the bank and its rule-based triggers.

Table 6: Failure-skill bank used by SkillCritic.

Failure skill	Trigger summary
Unverified uniformity	symmetry or evenly distributed language without an explicit invariance or universal justification
Unverified aggregation	total or summation language without a consistency or invariant check
Algebraic simplification errors	aggressive factorization, cancellation, or simplification patterns that commonly hide reasoning mistakes
Unaccounted complete depletion	use-up or exhaust reasoning without a residual-case analysis

For each segment s_k, the detector returns a risk value $c_k \in [0, 1]$. In the current implementation, the per-pattern severity is fixed at $\alpha = 0.7$, so the segment risk is the maximum severity among matched skills and is zero when no skill fires.

B.3 Skill-Aware Calibration

The calibration rule combines critic confidence and failure-skill activation. For segment k, let p_k denote the raw critic score and let c_k denote the aggregated failure-skill risk. The corrected actor-facing score is

$$\rho_k = \text{clip}(\lambda_t c_k g_k, 0, \rho_{\max}), \qquad u_k = p_k(1 - \rho_k),$$

where g_k is an over-optimism gate and $\rho_{\max} = 0.5$ bounds the shrinkage strength. In the main runs, the calibration scale uses $\eta = 0.08$ with a warmup from step $T_0 = 15$ to step $T_1 = 35$.

Only the top-$k = 4$ highest-score segments per trajectory are corrected. This keeps the correction local to the prefixes where over-optimistic critic values are most consequential.

Table 7: Core calibration hyperparameters.

Hyperparameter	Value	Role
Segment width	128 tokens	critic scoring granularity
α	0.7	per-pattern severity
top-k	4	number of corrected segments per trajectory
$\rho_{\max}$	0.5	max shrinkage strength
η	0.08	calibration scale
Warmup	steps 15–35	turns on correction progressively

B.4 Integration with Policy Optimization

The calibration is applied on the actor-facing reward path before downstream credit estimation. For a corrected segment, the actor-side signal is replaced by

$$\widetilde{r}_k = r_k(1 - \rho_k).$$

The resulting sequence is passed to the same GAE and PPO-style actor optimization used by the matched generative-critic baseline. The critic architecture, objective, and training target remain unchanged.

B.5 RandomSkill-B

RandomSkill-B keeps the detector pool size, top-k calibration, confidence gate, and reward-path integration unchanged, but replaces the semantic contents of the failure-skill regex library with frequency-matched semantically neutral triggers. It serves as a semantic-isolation control.

B.6 RLPT Critic Initialization

The RL-Post-Trained-Critic runs initialize the generative critic from a Stage-2 RL-pretrained checkpoint rather than from the SFT critic. In the archived runs, this initialization removes the earlier step-60 to step-80 instability window and makes the RLPT comparison stable enough to interpret.

C Additional Results

C.1 Full Benchmark Results

To keep the appendix readable, we split the full seven-benchmark comparison into the AIME-family block and the broader benchmark block rather than scaling a single ten-column table to the page width.

Table 8: AIME-family results at the best checkpoints. Entries are pass@1 (%).

Method	AIME24	AIME25	AIME26	AIME-Fam.	Avg7
SFT Critic					
Base	9.79	8.13	7.92	8.61	30.48
GRPO	12.92	11.87	8.13	10.97	33.64
PPO	12.92	14.17	8.75	11.95	36.24
GenAC-Style	12.04	11.83	8.29	10.72	35.17
SkillCritic	16.58	14.21	9.63	13.47	37.14
RL-Post-Trained Critic					
Base	9.79	8.13	7.92	8.61	30.48
GRPO	12.71	11.46	8.33	10.83	34.46
PPO	13.13	13.33	9.79	12.08	36.53
GenAC-Style	16.00	14.13	8.92	13.02	**37.77**
SkillCritic	**20.12**	13.57	9.42	**14.37**	37.71

Table 9: General-benchmark results at the same best checkpoints. Entries are pass@1 (%).

Method	AMC23	Math500	Olympiad	Minerva
SFT Critic				
Base	47.19	68.86	35.36	36.08
GRPO	52.34	73.50	38.75	37.96
PPO	57.81	77.22	41.99	40.81
GENAC-STYLE	54.28	76.30	41.91	41.51
SKILLCRITIC	57.56	78.18	42.22	41.61
RL-Post-Trained Critic				
Base	47.19	68.86	35.36	36.08
GRPO	55.94	74.19	39.19	39.38
PPO	56.72	78.55	42.87	41.29
GENAC-STYLE	**60.06**	**79.80**	**44.01**	41.49
SKILLCRITIC	57.72	78.15	43.31	**41.69**

Table 10: Compact summary of the main comparison. Best values are evaluated at each method's own best checkpoint.

Critic regime	Method	Best AIME-Fam.	Best Avg7	Note
SFT Critic	PPO	11.95	36.24	strongest plain PPO baseline
SFT Critic	GENAC-STYLE	10.72	35.17	generative-critic baseline
SFT Critic	SKILLCRITIC	**13.47**	**37.14**	strongest in SFT-critic group
RLPT Critic	PPO	12.08	36.53	plain RL baseline with RLPT critic
RLPT Critic	GENAC-STYLE-RLPT	13.02	**37.77**	strongest Avg7 in RLPT group
RLPT Critic	SKILLCRITIC-RLPT	**14.37**	37.71	strongest AIME-Fam. in RLPT group

The split tables keep the page layout stable while still preserving the benchmark-level picture. The compact summary in Table 10 highlights the two main takeaways: within the SFT-critic setting, SKILLCRITIC dominates both AIME-Fam. and Avg7; within the RLPT setting, SKILLCRITIC-RLPT yields the strongest AIME-family score while GENAC-STYLE-RLPT remains marginally higher on the broader Avg7 aggregate.

C.2 STEP-WISE RLPT RESULTS

The original appendix used four large heat-map tables for the checkpoint traces. Here we keep the cleaner presentation but add back more support through three compact views: (i) colored checkpoint curves, (ii) a best-checkpoint summary, and (iii) selected late-stage checkpoint tables. This retains the training dynamics without pushing dense heat-map matrices back into the appendix.

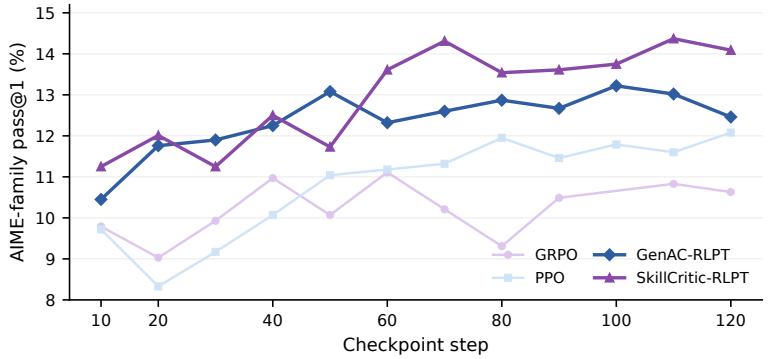

Figure 6: AIME-family pass@1 across RLPT training checkpoints. The matched RLPT methods are emphasized against GRPO and PPO.

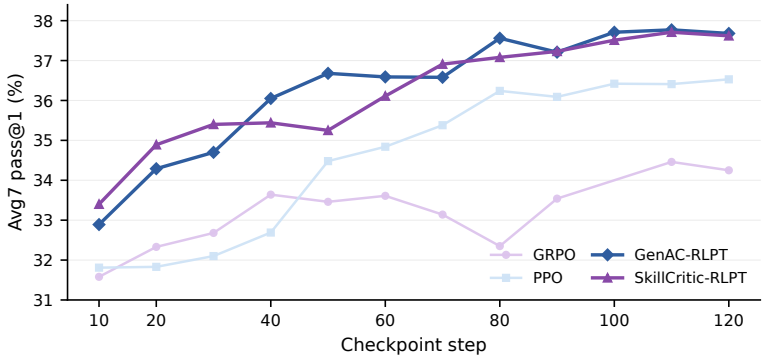

Figure 7: Avg7 pass@1 across RLPT training checkpoints.

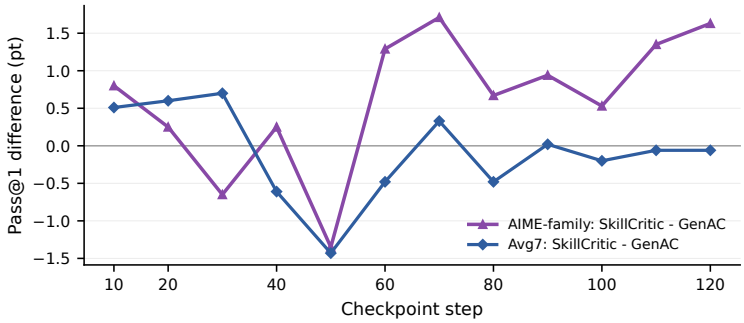

Figure 8: Matched RLPT comparison. Positive values indicate that SKILLCRITIC-RLPT outperforms GENAC-STYLE-RLPT at the same checkpoint. The gain is consistently larger on AIME-family than on Avg7.

The checkpoint curves show the same regime split as the main text, but with a little more detail restored. On AIME-family tasks, SKILLCRITIC-RLPT becomes clearly stronger after step 60 and reaches its strongest score at step 110. On Avg7, the two matched RLPT methods remain close from step 80 onward, with GENAC-STYLE-RLPT slightly stronger on the broader benchmark aggregate. Figure 8 makes this asymmetry explicit: the checkpoint-wise advantage of SKILLCRITIC is materially larger on AIME-family than on Avg7.

C.3 FAILURE-SKILL RELEVANCE

Table 14 reports the randomized-skill control using the AIME-family quantities most directly tied to the semantic-isolation claim. Table 15 then reports the hit-conditioned split.

RandomSkill-B removes most of the AIME-family advantage while leaving Avg7 close to the two matched RLPT runs. The hit-conditioned split localizes the gain further: the improvement is concentrated on the skill-hit subset and is small on the remaining majority subset.

C.4 CRITIC STABILITY

The main paper treats critic stability as a separate bottleneck from failure-skill calibration. The archived diagnostics support this with two observations.

First, critic estimates can drift away from realized outcomes. Near the unstable stage, the matched generative critic reaches a mean predicted value of 0.714 while the realized reward is only 0.062. The hard-veto line shows a similar discrepancy, with mean critic value 0.681 and realized reward 0.104. Second, the unstable hard-veto run shows a broad collapse in training dynamics: entropy rises sharply, the valid critic-output rate drops, and the average response length contracts substantially.

Table 11: Best-Avg7 checkpoint summary for the RLPT comparison.

Method	Step	AIME-Fam.	Avg7
GRPO	110	10.83	34.46
PPO	120	12.08	36.53
GENAC-STYLE-RLPT	110	13.02	**37.77**
SKILLCRITIC-RLPT	110	**14.37**	37.71

Table 12: Selected late-stage AIME-block checkpoints for the matched RLPT comparison.

Method	Step	AIME24	AIME25	AIME26	AIME-Fam.	Avg7
GENAC-STYLE-RLPT	60	14.75	11.83	10.38	12.32	36.59
GENAC-STYLE-RLPT	80	14.75	14.33	9.54	12.87	37.56
GENAC-STYLE-RLPT	100	16.42	14.75	8.50	13.22	37.71
GENAC-STYLE-RLPT	110	16.00	14.13	8.92	13.02	37.77
GENAC-STYLE-RLPT	120	14.96	13.50	8.92	12.46	37.68
SKILLCRITIC-RLPT	60	17.21	13.99	9.63	13.61	36.11
SKILLCRITIC-RLPT	80	15.96	13.99	10.67	13.54	37.08
SKILLCRITIC-RLPT	100	17.21	12.74	11.29	13.75	37.51
SKILLCRITIC-RLPT	110	20.12	13.57	9.42	**14.37**	37.71
SKILLCRITIC-RLPT	120	20.33	12.95	9.00	14.09	37.62

These diagnostics suggest that skill-aware calibration and critic stability should be treated as related but distinct concerns.

D PROMPT, SKILL BANK, AND CASES

D.1 CRITIC PROMPT

The generative critic is prompted with the problem text, the partial-solution prefix, and a short instruction block that asks it to estimate the chance of eventual success before emitting a strict terminal score. The template is structured as follows.

> A language model is solving a math problem. Your job is to evaluate its probability of success given the partial solution it has already generated.
>
> `<problem>{problem}</problem>`
> `<partial_solution>{partial_solution}</partial_solution>`
>
> Instructions: (1) evaluate the difficulty of the problem; (2) recover the model's capability from the partial solution; (3) identify progress, error, or confusion; (4) estimate the probability of success if the solution is continued; (5) output one integer score from 0 to 10 inside `<final><score>N</score></final>`.

The appendix reports the prompt structure rather than the full raw prompt string. All experiments use the same strict parser.

D.2 DETECTOR OVERVIEW

The detector library is intentionally small. Each failure skill is matched by a compact set of keyword or structural regexes, and the detector returns the strongest matched severity for the current segment. This keeps the intervention auditable and makes the semantic-control experiment straightforward to interpret.

Table 13: Late-stage average over checkpoints $\{80, 100, 110, 120\}$ in the matched RLPT comparison.

Method	Mean AIME-Fam.	Mean Avg7	Mean non-AIME score
GENAC-STYLE-RLPT	12.89	**37.68**	**56.27**
SKILLCRITIC-RLPT	**13.94**	37.48	55.14

Table 14: Randomized-skill control with the RL-Post-Trained Critic.

Method	AIME24	AIME25	AIME26	AIME-Fam.	Avg7
SKILLCRITIC	20.12	13.57	9.42	14.37	37.71
RandomSkill-B	16.38	14.08	8.96	13.14	37.61
GENAC-STYLE	16.00	14.13	8.92	13.02	37.77

D.3 RANDOMIZED TRIGGER POOL

The RandomSkill-B control keeps the same detector interface but replaces the semantic triggers with frequent semantically neutral regexes. The control therefore preserves trigger frequency and calibration mechanics while removing the correspondence between trigger semantics and the reasoning error being corrected.

D.4 CASE I: UNVERIFIED AGGREGATION

At the critical prefix, the solution aggregates intermediate counts before establishing that the counted cases are disjoint. The matched GENAC-STYLE-RLPT critic still assigns a high score, $p_k \approx 0.78$, and the rollout continues along the flawed branch. In SKILLCRITIC, the same prefix activates the unverified-aggregation skill. The resulting correction, $\rho_k \approx 0.32$, lowers the actor-facing score to $u_k \approx 0.53$.

After this correction, the continuation changes. The calibrated rollout checks the overlap condition before recomputing the total and reaches the correct final answer. The baseline rollout keeps the original aggregation and remains incorrect. This case matches the intended use of SKILLCRITIC: the critic is retained, but its locally over-optimistic signal is reduced at the prefix where the semantic error first appears.

D.5 CASE II: UNVERIFIED UNIFORMITY

At the critical prefix in a second example, the rollout treats the cases as uniform after a local simplification. The matched GENAC-STYLE-RLPT critic assigns a high-confidence score, $p_k \approx 0.81$, even though a later exceptional branch breaks the assumed symmetry. In SKILLCRITIC, the same prefix activates the unverified-uniformity skill and reduces the actor-facing score to approximately $u_k = 0.57$.

The corrected rollout then separates the exceptional branch from the regular cases and reaches the correct answer. The baseline rollout keeps the uniformity assumption and fails later in the derivation. As in Case I, the intervention is local: SKILLCRITIC changes the continuation selected after a semantically risky prefix rather than imposing a uniform penalty on the full trajectory.

D.6 SEED ROBUSTNESS

The main paper reports the seed-0 run because it is the only fully completed run for every method family, but the archived training plan also includes rollout-seed variants for both SKILLCRITIC-RLPT and GENAC-STYLE-RLPT. Across these seed variants, the expected variation remains small relative to the main AIME-family gap.

This spread is small compared with the main RLPT AIME-family gap between SKILLCRITIC-RLPT and the matched GENAC-STYLE-RLPT baseline, so the qualitative conclusion remains unchanged.

Table 15: Hit-conditioned AIME-family analysis.

Subset	Share (%)	N	SKILLCRITIC	GENAC-STYLE	Gap
Skill-hit	21.5	310	23.60	18.10	5.50
Non-hit	78.5	1130	11.90	11.60	0.30

Table 16: Archived critic-stability diagnostics near the unstable stage.

Quantity	Matched critic run	Hard-veto run
Mean predicted value	0.714	0.681
Realized reward	0.062	0.104
Valid critic-output rate	—	55.8%
Early valid-output rate	—	97.0%
Mean response length at instability	—	421
Early mean response length	—	630
Entropy near instability	—	3.11
Early entropy	—	0.8

Table 17: Seed-robustness summary from the archived training plan.

Quantity	Approximate spread
Avg7 run-to-run variation	± 0.3
AIME-family run-to-run variation	± 0.5