

EST-PRM: Stress-Testing Process Reward Models Before They Become Load-Bearing

Ibne Farabi Shihab¹ and Fariya Afrin² and Sanjeda Akter¹ and Anuj Sharma³

¹Department of Computer Science, Iowa State University

²Department of Computer Science, Kalinga Institute of Industrial Technology

³Department of Civil, Construction & Environmental Engineering, Iowa State University

ishihab@iastate.edu

Abstract

Process reward models (PRMs) are widely used in language-model training with dense step-level supervision. They assume PRM scores are stable proxies for step correctness under label-preserving transformations. These transformations change reasoning structure but preserve final answers. We argue this assumption is not well validated. Such transformations can change how PRM scores relate to correctness signals, leading to different failure modes across models. To address this gap, we introduce **EST-PRM**, a stress-testing framework for dense process rewards. It applies three transformations: (1) step inflation, (2) dependency-aware step reordering, and (3) confidence markers. A vulnerability decomposition is defined that separates reward inflation from loss of correctness sensitivity. Five PRM-style models are evaluated on 4,687 reasoning chains from MATH-500, GSM8K, and PRMBench. The results indicate clear differences in vulnerability patterns across models. Math-Shepherd shows the strongest sensitivity to the position-sensitivity attack, with a Pearson correlation drop of 0.139 ± 0.031 and a $29.7 \pm 4.2\%$ score inflation rate. Qwen2.5-Math-PRM is most affected by step inflation, reaching a $47.6 \pm 4.3\%$ inflation rate. Confidence-based perturbations also distort reward calibration, revealing inconsistencies in correctness estimation. Three mitigation strategies are evaluated, highlighting trade-offs between robustness coverage and false-positive rates.

1 Introduction

Process reward models (PRMs) are learned evaluators that assign scores to intermediate reasoning steps and are used to guide process-supervised search, candidate filtering, and optimization. Their usefulness depends not only on detecting incorrect steps on naturally occurring traces, but also on retaining correctness-aligned

scores when a policy or search procedure changes the form of a reasoning trace. This dense, step-level supervision enables credit assignment over extended reasoning trajectories, but it also creates a stronger robustness requirement than agreement with ground-truth labels on natural traces: PRM scores must remain aligned with correctness under distributional shifts induced by policy optimization over high-reward trajectories. Under such selection pressure, any systematic deviation between PRM scores and true correctness becomes exploitable and can introduce bias into both training and inference-time selection. EST-PRM studies this PRM-specific reliability question. Other systems based on outcome-level RLVR, teacher-student distillation, or routing motivate the broader importance of reliable optimization signals, but they do not necessarily use PRMs and are not evaluated in this paper.

We argue that standard evaluations of PRMs are insufficient to establish robustness under such conditions. Existing benchmarks measure agreement with human or automatically verified labels on naturally generated reasoning chains (Lightman et al., 2024; Wang et al., 2024). Recent reasoning-model evaluations further demonstrate the use of established benchmarks to assess the performance of long-form reasoning systems (He et al., 2025). While necessary, these evaluations do not test whether PRM scores remain stable under label-preserving structural transformations. This limitation is consistent with broader concerns about proxy objectives and reward misspecification in AI systems (Amodei et al., 2016; Manheim and Garrabrant, 2018). In RLHF, prior work has also identified systematic limitations and vulnerabilities in reward modeling (Casper et al., 2023). However, dense step-level supervision creates an additional PRM-specific surface for such failures, since each intermediate scoring decision becomes a potential point at which superficial structural

changes may affect the evaluation signal.

To interrogate this surface, we introduce three classes of label-preserving transformations that preserve final correctness while perturbing reasoning structure (Figure 1):

1. **Step-inflation** increases reasoning length through insertion of semantically redundant restatement or verification steps and alters the per-step reward distribution without affecting the final answer.
2. **Position-sensitivity** permutes steps under dependency constraints and exposes reliance on absolute step ordering rather than logical structure.
3. **Confidence-injection** augments steps with discourse markers (e.g., “clearly”, “by definition”) that remain semantically vacuous but may influence learned scoring heuristics.

Each transformation reflects a distinct hypothesis about structural failure modes in PRM scoring functions. The transformations are designed as targeted structural stress tests rather than as a comprehensive enumeration of adversarial attacks or an estimate of their prevalence under adaptive policy optimization. We additionally evaluate template-free naturalistic edits to assess whether these sensitivities extend beyond the controlled stress tests in Appendix J.

We conduct an empirical study on 4,687 retained reasoning chains across five PRM-style models with diverse architectures and training regimes: Math-Shepherd-7B (Wang et al., 2024), Qwen2.5-Math-PRM-7B (Zhang et al., 2025), RLHFlow PRM (RLHFlow Team, 2024), Skywork-o1-Open-PRM-8B (He et al., 2025), and DeepSeek-Math-7B-PRM (Shao et al., 2024). We also include a zero-shot Llama-3-8B-Instruct critic as a lightweight, format-insensitive baseline. Across models, we observe consistent but heterogeneous vulnerability patterns. Math-Shepherd and Skywork-PRM show strong sensitivity to step ordering, whereas Qwen2.5-Math-PRM and DeepSeek-PRM show stronger sensitivity to step-inflation. These results indicate that vulnerabilities vary across models and manifest as distinct dominant failure modes. Strong performance on natural traces does not therefore imply robustness under the structural perturbations tested here.

Our contributions include:

- A targeted structural stress-test suite for process reward models that evaluates robustness under label-preserving structural transformations and complements existing stress-testing frameworks for sparse reward models.
- A formal vulnerability framework that separates reward inflation effects from degradation of correctness-aligned signal quality under structural perturbations.
- A scalable experimental protocol over 1,000 problems (4,687 reasoning chains), including corrupted-chain validation and human verification to assess label preservation.
- A characterization of systematic, model-dependent failure modes across five PRM families, showing that robustness does not correlate straightforwardly with natural-input accuracy.
- An evaluation of mitigation strategies with analysis of trade-offs between detection sensitivity and false-positive rates under structurally perturbed reasoning traces.
- A release of the full benchmark suite, attack implementations, and human-validated annotations to support reproducibility and robustness evaluation of PRMs in downstream applications.

Our findings show that PRM robustness is sensitive to structural perturbations in reasoning traces, and that different models exhibit distinct dominant failure modes rather than uniform degradation. These results highlight the need for robustness evaluation beyond natural-input agreement and provide a foundation for more reliable evaluation of process-level supervision. We emphasize that EST-PRM is a targeted study of PRM-specific structural robustness: it does not claim to characterize the robustness of outcome-level RLVR, teacher–student distillation losses, routing objectives, or other reasoning optimization signals.

2 Related Work

Our work builds on recent advances in process reward models, reward-model evaluation, and reward hacking analysis for learned evaluators. We examine the robustness of dense step-level supervision under label-preserving transformations.

Below, we review the most closely related literature on PRMs while Appendix A presents a broader discussion of reward hacking and proxy gaming, calibration methods, reinforcement learning pipelines, and alignment evaluation frameworks.

Process reward models, benchmarks, and EST-PRM positioning. Process reward models (PRMs) were introduced to enable dense, step-level supervision for reasoning tasks. Early open-source formulations such as those of Lightman et al. (2024) and Wang et al. (2024) established the basic paradigm of process-level scoring. Subsequent higher-capacity variants, including Skywork-PRM (He et al., 2025) and Qwen2.5-Math-PRM (Zhang et al., 2025), improved coverage and performance on mathematical reasoning benchmarks. Existing evaluation suites for PRMs mainly focus on detection of naturally occurring or synthetically injected reasoning errors at the step level and achieve strong baseline reliability under distribution shifts and adversarial error conditions. EST-PRM stands as complementary to this line of work: rather than evaluating whether a PRM identifies incorrect steps already present in the data, it tests whether PRM scores remain invariant under label-preserving transformations that do not alter correctness. Table 1 summarizes the relationship between EST-PRM and prior evaluation methodologies.

3 The EST-PRM Framework

A process reward model is a function $\text{PRM}(q, s_1, \dots, s_k) \in \mathbb{R}$ that scores a reasoning chain $(s_1, \dots, s_k)$ for a question q . We write $x = (q, s_1, \dots, s_k)$ for a scored chain and $Y(x) \in \{0, 1\}$ for its ground-truth correctness label. A stress transformation $\mathcal{A}$ maps x to a modified chain $\mathcal{A}(x)$ while preserving both the final answer and the ground-truth label, so that $Y(\mathcal{A}(x)) = Y(x)$. Because Y is by construction unchanged, any movement of the PRM score from $S_P(x)$ to $S_P(\mathcal{A}(x))$ is a movement that cannot be justified by correctness, and is therefore the quantity we want to measure. Table 1 summarizes how EST-PRM differs from existing PRM evaluation methodologies. While reward hacking and superficial-feature reliance are established concerns in reward modeling, existing PRM benchmarks primarily emphasize error detection. EST-PRM instead evaluates whether PRM scores

Table 1: Comparison of PRM evaluation methodologies. EST-PRM evaluates PRM invariance and exploitability under correctness-preserving structural transformations, complementing existing benchmarks that primarily evaluate natural-error detection.

Work	Unit tested	Perturbation	PRM specific?	Dense attack?
ProcessBench	Process steps	Natural errors	Yes	No
PRMBench	Process steps	Step-level errors	Yes	No
RewardBench	Outcome reward	Diverse heuristic/LLM errors	No	No
RLHF Hacking	Outcome reward	Adversarial inputs, mostly outcome-level	No	No
EST-PRM (ours)	Process steps	Step inflation, position shifts, confidence perturbations	Yes	Yes

remain invariant under correctness-preserving structural transformations and whether such transformations can be exploited to inflate scores. Our contribution is therefore not the discovery of reward hacking itself, but a PRM-specific evaluation framework that combines dense step-level attacks, label-preserving transformations, separate measurements of reward inflation and correctness-signal degradation, cross-model vulnerability profiling, and human validation of attack validity.

EST-PRM measures two such movements. The first is reward inflation: the transformed chain receives a higher score even though its correctness label is unchanged. The second is correctness-signal degradation: across the benchmark, the correlation between PRM score and correctness weakens after the transformation is applied. These are not redundant. A PRM can inflate individual scores without losing population-level correlation (if inflation is consistent across both correct and incorrect chains), and it can lose correlation without any net inflation (if scores are merely shuffled within the existing range). Separating the two is the central diagnostic move of the framework.

Definition 1 (Correlation collapse). Let $B = \{(x_j, y_j)\}_{j=1}^n$ be a benchmark with nonconstant labels, and let $S_P(x)$ be the scalar score assigned by PRM P to chain x . The correlation collapse of P under transformation $\mathcal{A}$ on B is

$$\Delta\rho(P, \mathcal{A}, B) = \rho(S_P(X), Y) - \rho(S_P(\mathcal{A}(X)), Y), \quad (1)$$

where ρ is the Pearson correlation over the empirical distribution induced by B .

Definition 2 (Score inflation and score-inflation rate). The mean score inflation of P under $\mathcal{A}$ on

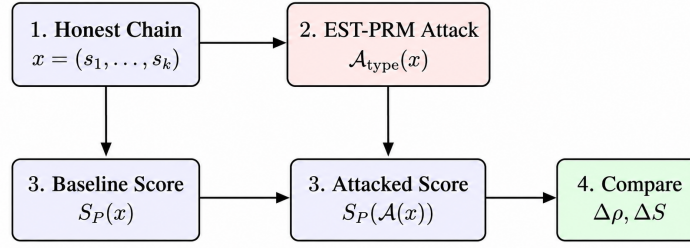


Figure 1: The EST-PRM pipeline. A question and its corresponding honest reasoning chain are subjected to a label-preserving attack transformation. Process reward model (PRM) scores are then evaluated before and after transformation to quantify correlation collapse and reward inflation, thereby isolating vulnerabilities arising from structurally induced reward gaming.

B is

$$\Delta S(P, \mathcal{A}, B) = \frac{1}{n} \sum_{j=1}^n [S_P(\mathcal{A}(x_j)) - S_P(x_j)]. \quad (2)$$

For a threshold $\tau > 0$, the score-inflation rate is

$$I_\tau(P, \mathcal{A}, B) = \frac{1}{n} \sum_{j=1}^n \mathbf{1} \left\{ S_P(\mathcal{A}(x_j)) > (1 + \tau) S_P(x_j) \right\}. \quad (3)$$

The main experiments use $\tau = 0.1$.

A concrete illustration of what each transformation does on a short example chain is given in Appendix G (Table 12), together with the full attack templates and the dependency-graph construction, strict eligibility algorithm, and rejection categories are provided in Appendix L.

3.1 Formal Vulnerability Analysis

The empirical patterns we report in Section 5 are not arbitrary observations; each attack class is designed to exploit a structural property of common PRM scoring schemes. We make this connection precise here by isolating two such properties and giving conditions under which each attack class can or cannot move the score.

Definition 3 (Mean-aggregated PRM). A PRM is mean-aggregated on a chain $(q, s_1, \dots, s_k)$ if its scalar chain score can be written as

$$S_P(q, s_1, \dots, s_k) = \frac{1}{k} \sum_{i=1}^k r_i, \quad (4)$$

where $r_i = \text{PRM}_i(q, s_1, \dots, s_i)$ is the reward assigned to the i th step under the model’s native scoring format.

Definition 4 (Position-invariant PRM). A PRM is position-invariant on a class of dependency-preserving permutations Π if

$$S_P(q, s_{\pi(1)}, \dots, s_{\pi(k)}) = S_P(q, s_1, \dots, s_k) \quad (5)$$

for every $\pi \in \Pi$. Failure of this condition means that at least one allowed permutation changes the score while preserving the multiset of reasoning steps and the final-answer label.

With these properties named, the effect of step-inflation on a mean-aggregated PRM follows directly from arithmetic.

Theorem 5 (Step-inflation under mean aggregation). Let P be mean-aggregated on a chain $(s_1, \dots, s_k)$ with average step reward $\bar{r}$. Suppose a step-inflation transformation inserts m additional restatement steps whose average reward under the same PRM is $\bar{r}_{\text{ins}}$. The transformed score is

$$S_P(\mathcal{A}_{\text{step}}(x)) = \frac{k\bar{r} + m\bar{r}_{\text{ins}}}{k + m}, \quad (6)$$

and the score change is

$$S_P(\mathcal{A}_{\text{step}}(x)) - S_P(x) = \frac{m}{k + m} (\bar{r}_{\text{ins}} - \bar{r}). \quad (7)$$

Thus step-inflation increases the mean-aggregated score if and only if $\bar{r}_{\text{ins}} > \bar{r}$.

Proof. The original score is $S_P(x) = \frac{1}{k} \sum_{i=1}^k r_i = \bar{r}$. After insertion, the numerator of the mean score is the sum of the

original k rewards and the inserted m rewards, giving $(k\bar{r} + m\bar{r}_{\text{ins}})/(k + m)$. Subtracting $\bar{r}$ yields $m(\bar{r}_{\text{ins}} - \bar{r})/(k + m)$. $\square$

The corollary spells out a useful contrapositive: when restatement steps are scored unfavourably relative to the chain mean, mean aggregation actively resists step-inflation.

Corollary 6 (Restatement penalty). *For a mean-aggregated PRM, step-inflation cannot produce positive mean score inflation on a chain whenever the inserted restatement steps receive lower average reward than the original steps.*

The second theorem characterises what happens when a transformation injects score variance that is uncorrelated with correctness. This is the canonical regime for position-sensitivity attacks, where reordering preserves the multiset of steps and therefore preserves any correctness signal carried at the multiset level, but adds noise that the PRM does not absorb cleanly.

Theorem 7 (Correlation degradation under correctness-independent perturbations). *Let Y be a nonconstant correctness label, let S be the honest PRM score with variance $\sigma_S^2 > 0$, and let $S' = S + \eta$ be the score after a label-preserving transformation. Suppose η is centred, uncorrelated with Y , and uncorrelated with S , with variance $\sigma_\eta^2 \geq 0$. Then*

$$\rho(S', Y) = \rho(S, Y) \frac{\sigma_S}{\sqrt{\sigma_S^2 + \sigma_\eta^2}}. \quad (8)$$

Consequently, when $\rho(S, Y) > 0$, the correlation collapse is

$$\Delta\rho = \rho(S, Y) \left(1 - \frac{\sigma_S}{\sqrt{\sigma_S^2 + \sigma_\eta^2}} \right), \quad (9)$$

which is nonnegative and is strictly positive whenever $\sigma_\eta^2 > 0$.

Proof. Because η is uncorrelated with Y , $\text{Cov}(S', Y) = \text{Cov}(S + \eta, Y) = \text{Cov}(S, Y)$. Because η is uncorrelated with S , $\text{Var}(S') = \text{Var}(S) + \text{Var}(\eta) = \sigma_S^2 + \sigma_\eta^2$. Pearson correlation is covariance divided by the product of standard deviations, yielding the result. $\square$

These two results give a clear theoretical lens for the experiments. Theorem 5 predicts that a mean-aggregated PRM whose restatement scores

hover near the chain mean will show step-inflation $\Delta\rho$ without large mean score change, while a PRM whose restatement scores exceed the chain mean will show both. Theorem 7 predicts that position attacks should produce correlation collapse roughly proportional to the variance of position-induced noise, with relatively little movement in the mean score. We test both predictions directly in Section 5.

4 Experimental Protocol

To provide a targeted structural stress test, we generate 5,000 reasoning chains spanning 1,000 distinct problems, drawn in equal proportion from MATH-500 (400), GSM8K (400), and PRMBench (200). After preprocessing, 4,687 chains are retained. Unless otherwise stated, all main attack metrics are computed on this retained set; the position analysis uses the 3,742 chains satisfying the strict position-eligibility criterion. The resulting candidate-pool accounting and eligibility breakdown for the downstream selection experiment are provided in Appendix C. We characterize these experiments as a targeted structural stress test rather than a comprehensive robustness evaluation: the transformations are controlled probes designed to isolate specific structural failure hypotheses. The scope is limited to these probes and the complementary template-free naturalistic clarity-edit experiment in Appendix J; adaptive, jointly optimized, multilingual, code, dialogue, and multimodal attacks remain outside the present coverage. Failure on the controlled probes, together with the template-free naturalistic-edit evidence, is sufficient to show that accuracy on natural, unmodified reasoning traces alone does not certify structural robustness. Conversely, passing the evaluated probes should not be interpreted as evidence of comprehensive robustness.

Chain Generation and Corrupted-Chain Validation For each problem, we generate five reasoning chains with **Meta-Llama-3-70B-Instruct** decoded at temperature $T = 0.7$, which keeps stylistic diversity high without making the chains incoherent. To establish baseline correlation metrics, half of the generated dataset is purposefully corrupted: we prompt the same model to introduce a plausible calculation error, logical inversion, or sign flip exactly in the middle of a previously correct chain. The corruptions are designed to be locally smooth, so that the surface style of

the chain does not change but the underlying reasoning does.

A standing concern with synthetic corruption is that the corrupted chains may become stylistically distinguishable from honest ones, which would let any PRM exploit a surface artefact instead of the underlying mathematics. We therefore validate the corruption pipeline. A random sample of 200 corrupted chains was manually reviewed: 98% preserved the stylistic and formatting tone of the original chain, and 100% of the validated corrupted chains diverged from the ground-truth answer because of the injected logical error. Invalid or ambiguous corruptions were excluded from the retained evaluation set, so that the benchmark tests whether PRMs detect mathematical faults rather than formatting artefacts.

Automatic Correctness Labeling Ground-truth binary correctness labels $Y \in \{0, 1\}$ are assigned automatically. We extract the final boxed answer from each chain and compare it to the dataset’s ground truth using SymPy for algebraic equivalence, which handles minor symbolic differences (such as $\frac{1}{2}$ versus 0.5) without overfitting to literal string match. The initial extraction failure rate across all generated chains was 2.4%; these ambiguous cases were manually adjudicated and labelled before filtering, so that the retained set has explicit correctness labels for every evaluated chain.

Evaluation Settings The PRM set comprises Math-Shepherd-7B, Qwen2.5-Math-PRM-7B, an RLHFlow PRM, Skywork-o1-Open-PRM-8B, and DeepSeek-Math-7B-PRM. The LLM-as-critic baseline uses Meta-Llama-3-8B-Instruct. The submitted baseline uses a scalar chain-level score; a dense stepwise critic with mean aggregation is evaluated separately in Appendix K as a granularity-matched control. We fix the random seed to 42 and compute 10,000 problem-cluster bootstrap resamples to report 95% confidence intervals across the primary metrics. We also include paired multiplicity controls for step inflation, confidence injection, and position sensitivity to distinguish structured vulnerabilities from pool expansion or surface-format effects.

For the primary analysis, the native PRM chain score is the arithmetic mean of the per-step scores.

We define the inflation statistic as

$$I_{0.1} = \frac{1}{n} \sum_j \mathbf{1}\{S(A(x_j)) > 1.1S(x_j)\},$$

while paired mean score change is reported separately. Additional aggregation rules, including minimum, maximum, final-step, and geometric mean, are evaluated separately as sensitivity analyses. Full scoring, inference, calibration, and implementation details are provided in Appendix M. Step inflation and confidence injection are evaluated on the retained 4,687-chain source pool. The revised position analysis uses a single pre-specified reward-front-loaded topological order on the 3,742 strictly eligible chains; the remaining 945 chains receive no position-transformed view. Additional aggregation rules, including minimum, maximum, final-step, and geometric mean, are evaluated separately as sensitivity analyses (Appendix H, Table 13). Full experimental details, including the prompts and implementation procedures, are provided in Appendices E and N.

Validation of Label Preservation The validity of our framework relies on the assumption that the applied transformations preserve correctness labels. We empirically validate this assumption through human annotation. Three expert annotators independently evaluated a stratified sample of 500 attacked chains across all three perturbation classes. Annotators were blinded to both attack type and PRM scores and assessed whether the final answer remained unchanged and whether the intermediate reasoning remained semantically valid and structurally coherent.

Because position reordering presents the greatest semantic risk, we additionally perform a position-only blind audit of 300 chains. On this audit, 94.7% preserved the final answer and 89.0% preserved the intermediate reasoning and local justification, corresponding to an estimated invalidity rate of 11.0%. Applying the strict eligibility rule yields a subset of 239 chains, for which answer preservation increases to 99.2% and reasoning preservation to 97.5%, with a residual invalidity rate of 2.5%. See Table 2 for the human validation results. The complete annotation rubric and adjudication criteria are provided in Appendix F.

The strict position procedure rejects transformations that violate dependency, discourse, coreference, scope, or local-justification constraints. The final answer step is kept fixed, and among cur-

Table 2: Human validation of label preservation for position transformations. “Answer preserved” indicates that the final mathematical answer remains unchanged, while “Reasoning preserved” indicates that the intermediate reasoning remains semantically valid and locally justified.

Validation	Answer pres.	Reasoning pres.	Invalid rate
Position-only blind audit ($n = 300$)	94.7%	89.0%	11.0%
Strict-eligibility subset ($n = 239$)	99.2%	97.5%	2.5%

rently available nodes in the dependency graph, the primary position transformation selects the node with the largest native reward, with ties broken by the original step index. This produces a fixed reward-front-loaded topological order. Reverse- topological and reward-back-loaded orders are retained only as sensitivity analyses.

Applying the strict eligibility rule to the complete source pool yields 3,742 chains with an admissible position transformation. The remaining 945 chains receive neither a position attack nor its paired multiplicity control. Thus, the revised position results do not claim invariance over arbitrary presentation orders; they measure sensitivity to the specified reward-front-loaded reorder under substantially stronger structural and semantic validity checks.

5 Empirical Evaluation

The empirical evaluation is conducted in two stages. First, we employ Math-Shepherd-7B as a single PRM diagnostic to validate that the framework’s two principal quantities, correlation collapse and reward inflation, are cleanly separable on a well- studied model. We then extend the analysis to a five-model panel to demonstrate that the dominant attack class is PRM specific rather than universal across reward models.

Math-Shepherd as a Diagnostic Probe We begin with Math-Shepherd-7B over the retained 4,687-chain dataset. The baseline point-biserial Pearson correlation between Math-Shepherd’s chain score and binary correctness is $\rho = 0.381 \pm 0.012$, indicating nontrivial but imperfect discrimination. Additional discrimination diagnostics, including AUROC, AUPRC, and balanced accuracy, are reported in Appendix I. EST-PRM therefore focuses on whether this existing

correctness-aligned signal remains stable under label-preserving structural transformations. Table 3 reports the correlation collapse and reward inflation outcomes.

Table 3: Math-Shepherd-7B on 4,687 retained chains. ρ_{base} is the point-biserial Pearson correlation between the mean native step score and binary chain correctness. $\Delta\rho = \rho_{\text{base}} - \rho_{\text{attack}}$, with positive values indicating loss of correctness-aligned signal. “Score inflation” denotes the paired mean score change, and “inflation rate” denotes the percentage of chains whose relative score increases by more than 10%. Uncertainty intervals are 95% problem-cluster bootstrap intervals.

Attack	$\Delta\rho$	Score infl.	Infl. rate
Step-infl.	$0.128 \pm .023$	$-0.027 \pm .009$	$22.4 \pm 3.1\%$
Position Sensitivity	$0.139 \pm .031$	$+0.018 \pm .006$	$29.7 \pm 4.2\%$
Confidence	$0.011 \pm .004$	$-0.003 \pm .001$	$6.5 \pm 1.2\%$

The pattern aligns with the diagnostics in Section 3.1. Step-inflation reaches $\Delta\rho = 0.128$ but has a slightly negative mean score change of -0.027 , which is the regime described by Corollary 6: the restatement steps receive marginally lower average reward than the original steps, so mean aggregation does not lift the overall score even though the correlation with correctness weakens. Position-sensitivity reaches $\Delta\rho = 0.139$ with a positive mean score change of $+0.018$ and a 29.7% inflation rate, which is the regime described by Theorem 7: the fixed reward-front-loaded topological reorder injects score variation that is not absorbed by correctness, so correlation drops and individual scores move. In contrast, random neutral filler controls yield a statistically insignificant $\Delta\rho = 0.012 \pm 0.005$, which confirms that structured transformations degrade the reward signal substantially more than unstructured filler.

Multi-PRM Vulnerability Profiles Two PRMs trained on different base models and with different objectives may inherit different structural properties, and the three attacks should reveal those differences. We therefore expand the evaluation to five process-reward-style models and the LLM-as-critic baseline. Table 4 reports the correlation collapse across models, Table 5 reports the matching score-inflation rates, and Figure 2 visualises the per-model profile.

The two tables and the figure tell a consistent story: the vulnerability profile is model-specific rather than universal. Math-Shepherd and Skywork-PRM share a dominant failure mode in position-sensitivity, where reordering steps causes

Table 4: Correlation collapse $\Delta\rho$ across five process-reward-style models, an LLM critic, and three attacks. Step-inflation and confidence-injection use the retained 4,687-chain benchmark, whereas the revised position analysis uses the 3,742-chain strict-eligibility subset. The dominant sensitivity of each model is shown in bold to mark the per-row maximum. 95% bootstrap confidence intervals are shown.

Model	ρ_{base}	Step Inflation	Position	Confidence
Math-Shepherd-7B	$0.381 \pm .012$	$+0.128 \pm .023$	$+0.139 \pm .031$	$+0.011 \pm .004$
Qwen2.5-Math-PRM-7B	$0.434 \pm .014$	$+0.089 \pm .017$	$+0.035 \pm .011$	$+0.015 \pm .005$
RLHFlow-PRM-8B	$0.100 \pm .011$	$-0.059 \pm .021$	$-0.014 \pm .009$	$+0.002 \pm .004$
Skywork-o1-Open-PRM	$0.412 \pm .015$	$+0.048 \pm .013$	$+0.115 \pm .029$	$+0.020 \pm .006$
DeepSeek-Math-PRM	$0.395 \pm .013$	$+0.093 \pm .020$	$+0.049 \pm .014$	$+0.010 \pm .003$
Llama-3-8B-Critic	$0.340 \pm .010$	$+0.010 \pm .005$	$+0.014 \pm .006$	$+0.006 \pm .002$

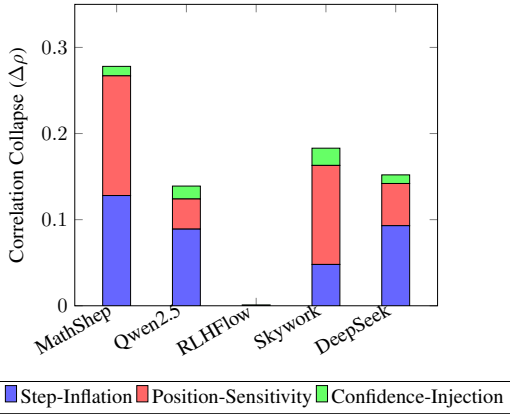


Figure 2: Model-by-attack profile of correlation collapse ($\Delta\rho$). Negative collapse (RLHFlow) is floored to zero for visualization. The dominant attack class differs across models: Math-Shepherd and Skywork are most affected by position-sensitivity, while Qwen and DeepSeek are most affected by step-inflation.

Table 5: Score-inflation rates (relative score increase $> 10\%$) showing model-specific exploitation risks. Per-row maxima are bolded.

Model	Step infl.	Pos infl.	Conf infl.
Math-Shepherd	$22.4 \pm 3.1\%$	$29.7 \pm 4.2\%$	$6.5 \pm 1.2\%$
Qwen2.5-PRM	$47.6 \pm 4.3\%$	$21.5 \pm 3.7\%$	$13.9 \pm 2.1\%$
RLHFlow-PRM	$1.9 \pm 0.7\%$	$1.4 \pm 0.6\%$	$2.8 \pm 1.0\%$
Skywork-PRM	$11.8 \pm 2.4\%$	$30.2 \pm 4.7\%$	$9.5 \pm 1.8\%$
DeepSeek-PRM	$38.8 \pm 3.9\%$	$15.1 \pm 2.8\%$	$8.1 \pm 1.4\%$

both the largest correlation drop and the highest inflation rate, suggesting that their per-step rewards mix step content with absolute step position more than the other models. The corresponding dataset-level breakdown of these dominant vulnerability modes, together with the exact post-filter sample counts, is reported in Appendix D. Qwen2.5-Math-PRM and DeepSeek-PRM, by contrast, are most exposed to step-count manipulations under the tested aggregation, with Qwen2.5-Math-PRM showing a striking 47.6% inflation rate under step-inflation. This pattern of architecture-specific spu-

rious correlations between PRM scores and superficial features of the reasoning chain is consistent with broader observations of spurious-feature reliance in reward models and with mechanistic studies of how reward models encode position-related signals. Prior work has shown that continual adaptation can induce unintended or spurious dependencies in language models (Zheng et al., 2025), while mechanistic analyses of continual fine-tuning have further linked such behavioral changes to alterations in the model’s internal representations (Lundstrom-Imanov, 2026). RLHFlow-PRM shows negative or near-zero collapse on all attacks, but its baseline correlation is also very low ($\rho_{\text{base}} = 0.100$), so the result mostly reflects that there is little signal there to degrade. The scalar LLM-as-critic control exhibits substantially smaller correlation collapse than the most vulnerable PRMs. The dense stepwise critic in Appendix K likewise shows substantially smaller collapse. We interpret these results as a control comparison, not evidence that zero-shot critics are generally preferable to PRMs. Additional visual analyses of attack effects, influence rates, model-specific robustness patterns, score inflation, mitigation behavior, and dataset filtering are provided in Appendix O.

A complementary view of the same effect appears in the score-distribution plot of Figure 3. Before the attack, correct and corrupted chains form well-separated modes; after the position attack, the corrupted distribution drifts toward higher scores while the correct distribution remains broadly anchored, and the two populations begin to overlap. The visual pattern is exactly what Theorem 7 predicts when a correctness-independent noise term is added to the score: covariance with Y is preserved, variance grows, and correlation shrinks.

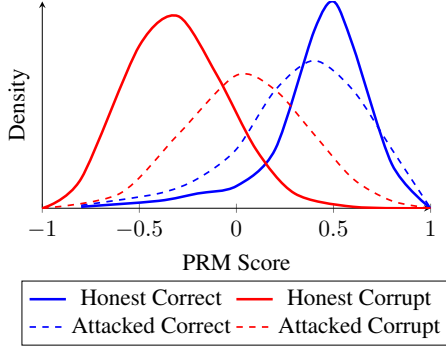


Figure 3: Empirical kernel-density estimates of Math-Shepherd score distributions for correct and corrupted chains before and after position-sensitivity attacks. Coordinates are exported from the released plotting script rather than hand-specified schematic curves. The attack blurs the separation between correct and corrupted chains by moving corrupted-chain scores toward the positive-reward region.

6 Mitigations

The vulnerability profiles in the previous section motivate two lightweight defenses targeting complementary aspects of the dense-reward attack surface. We evaluate them using five-fold problem-level cross-fitting on the Math-Shepherd PRM, with thresholds calibrated on training folds and evaluated on held-out problems. Throughout this section, $S_P(x)$ denotes the primary PRM score and $\mathcal{A}(x)$ a transformed chain.

The first defense, view-consistency flagging, detects disagreement between the source and transformed views. The second, a hybrid detector, combines PRM-critic disagreement, view inconsistency, and the residual from a score-versus-step-count fit. The PRM-critic combination draws on the reward-ensembling intuition of (Ramé et al., 2024). Thresholds are selected within each training fold to maintain an honest-chain false-positive rate of at most 5%.

The view-consistency detector achieves an AU-ROC of 0.78 ± 0.02 , while the hybrid detector reaches 0.81 ± 0.02 . In downstream selection, the hybrid recovers Math-Shepherd attacked-pool accuracy from 61.6% to 66.2%, compared with a clean source-only accuracy of 68.4%. These results provide a useful mitigation baseline, but do not constitute a complete solution; deployment-grade robustness likely requires robustness-aware PRM training rather than post hoc thresholding alone. Further details and full results are provided in Appendix B.

7 Conclusion

Process reward models are becoming load-bearing in language-model training, and the framework that depends on them rests on a trustworthiness assumption that should be tested under label-preserving transformations rather than assumed. The EST-PRM evaluation across 4,687 retained chains and five public process-reward-style models reveals PRM-specific vulnerability profiles: Math-Shepherd and Skywork-PRM are dominantly sensitive to position-sensitivity, while Qwen2.5-Math-PRM and DeepSeek-PRM are dominantly sensitive to step-inflation. The same PRM that wins on natural-input accuracy can therefore lose on operational robustness, and the loss is patterned rather than random.

These findings suggest that PRM deployment should treat label-preserving robustness as a first-class evaluation objective rather than relying only on natural-trace agreement.

Limitations

Our study has some limitations. First, the attack strategies considered in EST-PRM are intentionally simple and do not involve adaptive or jointly optimized policies. Consequently, more sophisticated adversaries that co-optimize transformations with reward maximization may expose additional failure modes beyond those identified in this work.

Second, our evaluation is restricted to mathematical reasoning tasks. While this setting is standard for process reward models, it remains unclear whether the observed vulnerability profiles generalize to other domains, including code generation, dialogue reasoning, or multimodal tasks.

Third, human validation is limited to 500 sampled chains. Although this provides evidence of systematic degradation, particularly under position-sensitivity attacks (Table 2), a larger-scale annotation study would improve the reliability and statistical strength of these findings.

Finally, the mitigation strategies considered in this work are simple and stationary, and do not incorporate learned or adaptive defense mechanisms that may offer stronger robustness through dynamic responses to structural perturbations.

These limitations leave open the need for stronger adaptive attack formulations, broader evaluation beyond mathematical reasoning, and learned or model-aware defenses that explicitly account for label-preserving transformations.

Societal Impact and Who May Be Harmed

A PRM that systematically rewards redundant, overconfident, or misleadingly ordered reasoning can amplify incorrect chains when used for educational feedback, scientific-assistance ranking, or safety-critical decision support. The immediate risk is not that EST-PRM attacks alter a correct answer; it is that a learned evaluator may preferentially select or reinforce an incorrect chain because of superficial presentation. Users most affected are those who rely on the evaluator’s ranking without independently verifying the underlying reasoning. EST-PRM is an evaluation tool and should not be interpreted as certifying a PRM for high-stakes deployment.

References

- Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. 2026a. [Anytime-valid answer sufficiency certificates for LLM generation via sequential information lift](#). *Preprint*, arXiv:2510.06478.
- Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. 2026b. Detecting proxy gaming in RL and LLM alignment via evaluator stress tests. In *Findings of the Association for Computational Linguistics: ACL 2026*.
- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
- Anastasios N Angelopoulos and Stephen Bates. 2023. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Foundations and Trends in Machine Learning*.
- Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candes, Michael I Jordan, and Lihua Lei. 2022. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*.
- Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. 2024. Conformal risk control. In *International Conference on Learning Representations*.
- Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. 2021. [Distribution-free, risk-controlling prediction sets](#). *J. ACM*, 68(6).
- Thomas Kleine Buening, Jonas Hübottter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. 2026. [Aligning language models from user interactions](#). *Preprint*, arXiv:2603.12273.
- Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. *International Conference on Machine Learning*.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, and 13 others. 2023. [Open problems and fundamental limitations of reinforcement learning from human feedback](#). *Preprint*, arXiv:2307.15217.
- Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformer: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*.
- Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of LLM inference using lookahead decoding. *International Conference on Machine Learning*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.
- Jujie He, Jiakai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. [Skywork open reasoner 1 technical report](#). *Preprint*, arXiv:2505.22312.
- Jonas Hübottter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. 2026. [Reinforcement learning via self-distillation](#). *Preprint*, arXiv:2601.20802.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of experts](#). *Preprint*, arXiv:2401.04088.
- Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. 2026. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*.

- Bhawesh Kumar, Charles Lu, Gauri Gupta, Anil Palepu, David R Bellamy, Ramesh Raskar, and Andrew Beam. 2023. Conformal prediction with large language models for multi-choice question answering. In *ICML Workshop on Neural Conversational AI*.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. [Rewardbench: Evaluating reward models for language modeling](#). *Preprint*, arXiv:2403.13787.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. *International Conference on Machine Learning*.
- Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. 2026. [Unifying group-relative and self-distillation policy optimization via sample routing](#). *Preprint*, arXiv:2604.02288.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. EAGLE: Speculative sampling requires rethinking feature uncertainty. *International Conference on Machine Learning*.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. *International Conference on Learning Representations*.
- G. O. Y. L. Lundstrom-Imanov. 2026. Mechanistic analysis of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2601.18699*.
- David Manheim and Scott Garrabrant. 2018. Categorizing variants of Goodhart’s law. *arXiv preprint arXiv:1803.04585*.
- Christopher Mohri and Tatsunori Hashimoto. 2024. Language models with conformal factuality guarantees. In *International Conference on Machine Learning*.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.
- John Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*.
- Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S Jaakkola, and Regina Barzilay. 2024. Conformal language modeling. In *International Conference on Learning Representations*.
- Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*.
- RLHFlow Team. 2024. RLHFlow: On-line RLHF pipeline with PRM variants. Open-source release on Hugging Face Hub: huggingface.co/RLHFlow/Llama3.1-8B-PRM-Deepseek-Data.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *Advances in Neural Information Processing Systems*.
- Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer.
- Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. *Annual Meeting of the Association for Computational Linguistics*.
- Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. 2026. OpenClaw-RL: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*.
- Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. 2026a. [Self-distilled rlvr](#). *Preprint*, arXiv:2604.03128.
- Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026b. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*.
- Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2026c. [Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation](#). *Preprint*, arXiv:2603.19220.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. [The lessons of developing process reward models in mathematical reasoning](#). *Preprint*, arXiv:2501.07301.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. *International Conference on Learning Representations*.

A Extended Related Work

Beyond direct PRM evaluation, related work also studies proxy-gaming analyses, the reliability of learned evaluators through calibration methods, reinforcement learning pipelines that depend on intermediate reward signals, and reasoning optimization and alternative supervision signals.

Reasoning optimization and alternative supervision signals. A growing set of recent reasoning-optimization systems explores alternative ways of transforming sparse correctness signals into stronger training and selection signals. These include SDPO (Hübotter et al., 2026), GOPD (Yang et al., 2026b), REOPOLD (Ko et al., 2026), Nemotron-Cascade 2 (Yang et al., 2026c), OpenClaw-RL (Wang et al., 2026), RLSD (Yang et al., 2026a), and SRPO (Li et al., 2026), which explore combinations of dense feedback, self-distillation, sample routing, and reward-based filtering. These approaches have demonstrated strong performance in challenging reasoning settings. For instance, Nemotron-Cascade 2’s 30B mixture-of-experts model (Fedus et al., 2022; Jiang et al., 2024) achieves gold-medal performance on the International Mathematical Olympiad and the International Olympiad in Informatics while activating only 3B parameters (Yang et al., 2026c), while REOPOLD shows that a 7B student can match a 32B teacher while achieving $3.3\times$ faster inference (Ko et al., 2026). Although these systems share the broader goal of improving the reliability and effectiveness of reasoning optimization signals, they differ substantially in their training and evaluation mechanisms and do not necessarily employ process reward models. These methods therefore provide broader context for the importance of reliable optimization signals, but are not directly evaluated by EST-PRM.

Reward hacking and proxy gaming. Reward hacking and proxy gaming have been studied extensively in learned reward functions and alignment objectives (Amodei et al., 2016; Manheim

and Garrabrant, 2018; Casper et al., 2023; Akter et al., 2026b). Benchmarking frameworks such as RewardBench (Lambert et al., 2024) test susceptibility to formatting artifacts, heuristic exploitation, and stylistic shortcuts at the level of scalar outcome rewards. These approaches rely on whole-response evaluation, where a single aggregated score or judgment applies to the full output. In contrast, PRMs define a structured scoring regime over intermediate reasoning steps, which yields a more fine-grained but also more fragile evaluation surface. EST-PRM extends proxy-gaming analysis to this dense-reward setting. It fixes final correctness and varies reasoning structure to isolate sensitivity of step-level scoring functions.

Calibration and uncertainty estimation. Calibration-based analyses of model confidence provide a related but distinct perspective on reward reliability. Classical calibration techniques such as temperature scaling and Platt scaling (Platt, 1999; Guo et al., 2017) measure misalignment between predicted probabilities and empirical correctness. More recent conformal prediction and uncertainty quantification methods for language models (Quach et al., 2024; Mohri and Hashimoto, 2024; Kumar et al., 2023; Akter et al., 2026a) extend this framework to distribution-free uncertainty estimation. These methods focus on predictive reliability under natural data variation and do not address invariance under structure-preserving transformations. EST-PRM instead evaluates whether the scoring signal itself allows systematic manipulation without change in semantic correctness, which exposes a complementary failure mode not captured by calibration-based analyses.

RL pipelines, learned evaluators, and decoding. A broader line of work studies reinforcement learning pipelines with learned or hybrid reward signals, including self-distillation and sparse-reward augmentation frameworks (Hübotter et al., 2026; Wang et al., 2026; Ko et al., 2026; Yang et al., 2026a; Li et al., 2026; Buening et al., 2026), which build on the RLHF paradigm (Ouyang et al., 2022). These approaches should be distinguished from process reward modeling: PRMs assign correctness-oriented scores to intermediate reasoning steps, whereas RLVR typically uses outcome-level verifiable rewards and distillation methods optimize teacher–student agreement or related distribution-matching objectives. Conse-

quently, the robustness question studied by EST-PRM is specific to PRM-style intermediate-step evaluation and does not directly extend to RLVR rewards or teacher–student distillation signals.

Test-time strategies such as process-supervised search (Snell et al., 2024) and verification-oriented decoding schemes (Leviathan et al., 2023; Chen et al., 2023; Li et al., 2024; Cai et al., 2024; Fu et al., 2024) also depend on the reliability of intermediate scoring or verification signals. EST-PRM focuses on this narrower setting: whether a PRM’s correctness-aligned signal remains stable under label-preserving structural transformations of a reasoning trace. This complements existing work on natural-trace accuracy, calibration, and reward-model reliability by testing structural robustness rather than introducing a general robustness framework for RLVR or distillation.

B Mitigation details and full results

Mitigation protocols We evaluate five mitigation approaches on Math-Shepherd. Three approaches use direct post-hoc scoring rules, while two additional detectors use a common five-fold problem-level cross-fitting protocol. Because these approaches use different fitting and thresholding procedures, we report the two groups separately.

Post-hoc mitigation methods We first consider three simple post-hoc defenses.

- **Ensemble flagging.** The first defense couples the primary PRM with a lightweight LLM critic using the conservative score

$$S_{\text{ens}}(x) = \min\{S_P(x), S_{\text{critic}}(x)\}.$$

A transformed chain $\mathcal{A}(x)$ is flagged when

$$S_{\text{ens}}(\mathcal{A}(x)) > (1 + \tau)S_P(x).$$

- **Conformal abstention.** The second defense follows the split-conformal construction (Vovk et al., 2005; Angelopoulos and Bates, 2023; Angelopoulos et al., 2024) together with the Learn-Then-Test framework (Angelopoulos et al., 2022; Bates et al., 2021). We compute non-conformity scores

$$E_j = -S_P(x_j)$$

on the calibration set and let q denote the $(1 - \alpha)$ empirical quantile of the calibration

scores. A transformed chain is abstained on when

$$-S_P(\mathcal{A}(x)) > q.$$

- **Length-normalized flagging.** The third defense defines

$$S_{\text{LN}}(x) = S_P(x) - \lambda|x|,$$

where $|x|$ is the number of steps. A transformed chain is flagged when

$$S_{\text{LN}}(\mathcal{A}(x)) > (1 + \tau)S_{\text{LN}}(x).$$

Cross-fitted mitigation methods We additionally evaluate view consistency and a hybrid detector under a common five-fold problem-level cross-fitting protocol. Each fold uses 800 problems to estimate detector statistics and the decision threshold and evaluates on the remaining 200 problems. All chains and views derived from a given problem remain in the same fold. No attacked view is used for fitting or threshold selection.

Within each training fold, the decision threshold is chosen as the smallest threshold whose empirical false-positive rate on honest chains is at most 5%. Thus, threshold selection uses only honest training-fold examples and targets the prescribed FPR level from below.

Each source chain has up to three views: the original source view, one step-preserving paraphrase, and one formatting-only view. The formatting-only view changes presentation while preserving lexical content, step count, step order, equations, numerical values, and final answer. A paraphrase that fails validation is not silently regenerated; the missing-paraphrase condition is recorded and the detector uses the available views. An identical paraphrase is retained and contributes zero paraphrase-induced variation.

As shown in Table 6, the primary structural stress tests produce substantially larger correlation collapse and inflation than the corresponding formatting, paraphrase, and neutral-text controls. For downstream selection, attack and control views are multiplicity matched. The position control is used to account for formatting and pool-expansion effects and is not interpreted as a non-adversarial reordering baseline.

View-consistency flagging. Let $\mathcal{V}(x)$ denote the available views of a source chain and let $S_P(v)$ denote the primary PRM score for view v . We define

Table 6: Math-Shepherd control and stress-test comparison. All values in this table are computed for Math-Shepherd-7B and are not reused for Qwen2.5 or other scorers. We report correlation collapse ($\Delta\rho$) and the inflation rate (relative score increase $> 10\%$). The formatting and paraphrase conditions are label-preserving controls; the remaining rows are the evaluated structural transformations.

Condition	$\Delta\rho$	Inflation rate
Whitespace/step-number formatting	0.003 ± 0.002	$2.1 \pm 0.6\%$
Scorer-specific semantic paraphrase	0.009 ± 0.004	$4.8 \pm 1.0\%$
Neutral filler	0.012 ± 0.005	—
Confidence injection	0.011 ± 0.004	$6.5 \pm 1.2\%$
Step inflation	0.128 ± 0.023	$22.4 \pm 3.1\%$
Fixed strict-subset position attack	0.139 ± 0.031	$29.7 \pm 4.2\%$

view inconsistency as the population variance of the available scores,

$$V(x) = \frac{1}{|\mathcal{V}(x)|} \sum_{v \in \mathcal{V}(x)} (S_P(v) - \bar{S}_P(x))^2,$$

where

$$\bar{S}_P(x) = \frac{1}{|\mathcal{V}(x)|} \sum_{v \in \mathcal{V}(x)} S_P(v).$$

A transformed view is flagged when its cross-fitted view-consistency score exceeds the corresponding training-fold threshold.

Hybrid detector. The hybrid detector equally combines three training-fold-standardized quantities: absolute PRM–critic disagreement, view inconsistency, and the residual from an ordinary-least-squares fit of PRM score versus step count. Let the corresponding standardized quantities be $D_z(x)$, $V_z(x)$, and $R_z(x)$. The hybrid score is

$$H(x) = \frac{1}{3} (D_z(x) + V_z(x) + R_z(x)).$$

The resulting score is thresholded using the same training-fold procedure targeting an honest-chain FPR of at most 5%.

Table 7: Post-hoc mitigation results on Math-Shepherd. We report true positive rate (TPR) for attack detection, false positive rate (FPR) on honest chains, and overall AUROC.

Mitigation	Step (TPR/FPR)	Pos (TPR/FPR)	AUROC
Ensemble flagging	0.19/0.05	0.34/0.06	$0.72 \pm .02$
Conformal abstention	0.00/0.01	0.00/0.01	$0.53 \pm .01$
Length-normalized flagging	0.13/0.04	0.22/0.05	$0.63 \pm .02$

Full quantitative results The post-hoc defenses provide limited protection. Ensemble flagging

has the highest AUROC among these methods, but its absolute detection rates remain modest. Conformal abstention has near-zero detection in this setting because attacked scores remain within the calibration support, while length normalization does not directly address structural transformations whose score changes are not explained by step count. The corresponding post-hoc mitigation results are reported in Table 7.

Under the common cross-fitted protocol, view consistency achieves an AUROC of 0.78 ± 0.02 , while the hybrid detector achieves 0.81 ± 0.02 . The hybrid detector also has higher reported TPR for both evaluated attack families while maintaining an honest-chain FPR near the prescribed 5% target. These results indicate that combining complementary signals provides a useful mitigation baseline, but detection remains incomplete. The complete cross-fitted mitigation results, including attacked-pool accuracy, are reported in Table 8.

Effect on downstream selection The downstream selection experiment uses the same retained source pool across clean, attack, and paired-control conditions. Detailed candidate-pool accounting, family-specific eligibility rules, accepted and rejected transformation counts, attack-family selection, label inheritance, and bootstrap procedures are reported separately in Appendix C.

For mitigation, the detector is applied only to added transformed views. Source candidates are never removed or rescored. Consequently, the clean source-only accuracy remains fixed at 68.4% by construction.

Under the Math-Shepherd fixed-position attack, the unmitigated attacked pool has accuracy 61.6%, compared with 68.4% for the clean source-only pool. View-consistency flagging raises attacked-

Table 8: Five-fold problem-level cross-fitted mitigation results on Math-Shepherd. Each fold uses 800 training problems to estimate detector statistics and the decision threshold and 200 held-out problems for evaluation. No attacked view is used for fitting or threshold selection. The threshold targets an honest-chain FPR of at most 5%. Attacked-pool accuracy is measured after applying the detector only to added transformed views; source candidates are never removed or rescored.

Mitigation	Step (TPR/FPR)	Pos (TPR/FPR)	AUROC	Attacked-pool accuracy
View consistency	0.36/0.05	0.48/0.05	$0.78 \pm .02$	65.1%
Hybrid detector	0.43/0.05	0.57/0.06	$0.81 \pm .02$	66.2%

pool accuracy to 65.1%, while the hybrid detector raises it to 66.2%. Thus, the hybrid recovers 4.6 of the 6.8 percentage points lost under the attack, while remaining 2.2 points below the clean source-only baseline.

This recovery should be interpreted as partial mitigation rather than complete robustness. Because source candidates are preserved throughout, the improvement comes from reducing the ability of added transformed views to affect selection rather than from changing the underlying source-chain scores.

Interpretation and limitations The mitigation experiments support a limited conclusion. The evaluated post-hoc defenses detect a nontrivial fraction of the studied transformed views, and the cross-fitted hybrid detector provides a useful baseline for combining complementary detection signals. However, substantial attack success remains, and the hybrid detector does not recover the full clean accuracy.

We therefore do not present these experiments as a complete defense or as evidence of deployment-grade robustness. Instead, they show that the studied vulnerability is partially detectable through presentation consistency, PRM-critic disagreement, and score-versus-length residuals, while also illustrating the limitations of post-hoc thresholding. Robustness-aware PRM training and other model-level interventions remain important directions for future work.

Finally, the downstream selection experiment should not be interpreted as evidence of degradation under full policy optimization or adaptive reinforcement learning. It is a controlled fixed-pool simulation showing that correctness-preserving structural transformations can alter PRM-guided selection sufficiently to reduce answer accuracy.

C Downstream selection pool and eligibility

The downstream selection experiment uses the same 1,000 retained problems and 4,687 source candidates across clean, attack, and paired-control conditions. Attack and control views are added only when both members of the pair pass the corresponding family-specific eligibility checks. A failed or ineligible view therefore adds neither candidate, while the original source candidate remains in every condition. The candidate-pool accounting for the downstream selection experiment is reported in Table 9.

Table 9: Candidate-pool accounting for the downstream selection experiment.

Pool quantity	Step inflation	Position	Confidence
Problems retained	1,000	1,000	1,000
Source candidates retained in every condition	4,687	4,687	4,687
Attack views added	4,687	3,742	4,687
Paired multiplicity-control views added	4,687	3,742	4,687
Source candidates receiving neither extra view	0	945	0

The transformation ledger records the eligibility status of each candidate view and the corresponding family-specific failure reason. Step-inflation uses one deterministic payload-generation attempt per eligible boundary, with no regeneration after failure. The saved ledger contains 4,687 accepted step-inflation pairs and no rejected pairs. Confidence injection produces one paired view for all 4,687 source chains. Strict position produces 3,742 admissible attack/control pairs; the remaining 945 source chains receive neither extra view.

For models with a statistically supported positive correlation collapse, the highlighted attack family is selected before downstream evaluation as the largest positive value in the submitted stress-test table. Negative values indicate improved correlation and are not treated as vulnerabilities. RL-

HFlow has no statistically reliable positive collapse; confidence injection is included only as a near-null comparator because it has the largest positive point estimate. Selection accuracy is never used to choose the attack.

A transformed view inherits the source chain’s answer label. For position, this inheritance is interpreted together with the 2.5% residual invalidity estimate rather than presented as a perfect semantic guarantee. All selection intervals use 10,000 paired problem-cluster bootstrap replicates.

D Per-Dataset Vulnerability Breakdown

Table 10 reports the exact sample counts after removing invalid or non-parsable chains, and Table 11 splits the performance of the dominant vulnerability modes by source dataset. Step-inflation remains stronger on GSM8K, where the average initial reasoning chain is shorter and inserted steps therefore carry more proportional weight. Position-sensitivity is particularly potent on MATH-500, where deeper reasoning graphs offer more valid permutations that structurally confuse the reward model.

Table 10: Sample counts per dataset after filtering invalid or non-parsable chains. The main experiments use the 4,687 retained chains.

Split	MATH-500	GSM8K	PRMBench
Initial	2,000	2,000	1,000
Post-filter	1,812	1,930	945

Table 11: Correlation collapse ($\Delta\rho$) by dataset split. Reported for the dominant attack class per PRM.

PRM (Dominant Attack)	MATH-500	GSM8K	PRMBench
Math-Shep (Position)	0.165 $\pm$.042	0.138 $\pm$.034	0.150 $\pm$.040
Qwen2.5 (Step-Infl)	0.065 $\pm$.015	0.108 $\pm$.021	0.095 $\pm$.019

E Experimental Prompts

We used three prompt templates: one to generate initial chains, one to corrupt them with a controlled logical error, and one to query the LLM-as-critic baseline. The chain-generation prompt asked the model to “solve the following math problem step-by-step, break your reasoning into distinct sentences or short mathematical steps, and end with your final answer enclosed in `\boxed{}`”, followed by the question. The corruption prompt asked the model to “modify exactly one step near

the middle of the chain to introduce a subtle logical error, arithmetic mistake, or sign flip, ensure the subsequent steps logically follow from this new incorrect step, leading to a wrong final answer, and preserve the exact style, formatting, and tone of the original chain”, followed by the chain. The LLM-critic prompt asked the model to “act as a strict mathematical reviewer”, to “evaluate the following reasoning chain step-by-step”, and to “output ‘Score: 1.0’ if all steps are correct and ‘Score: 0.0’ if there is a logical or arithmetic flaw anywhere in the chain”, followed by the chain. The exact strings used in the experiments are included verbatim in the artifact release.

F Annotation Guidelines

Expert annotators were given a three-part rubric during the human validation phase. They first checked answer preservation, judging whether the final boxed answer in the modified chain algebraically matched the original chain. They then checked reasoning preservation, ensuring that no new mathematical errors had been introduced by the modification; for step-inflation they verified that any restatement was factually aligned with prior steps, and for position-sensitivity they verified that the reordering did not violate causal mathematical dependencies (for example, by using a variable before it was defined). Finally, any chain receiving a negative judgment in either category from a majority of the annotators was flagged as an invalid attack and removed from the benchmark. Annotators were blinded to attack type and PRM scores throughout.

G Attack Templates and Configuration

Table 12 shows what each transformation does on a concrete short example chain. The position attack illustrates a valid reordering of independent steps while preserving the final answer. Position sensitivity is evaluated using a single pre-specified reward-front-loaded topological ordering subject to the dependency constraints described below.

Step-inflation attacks insert template restatement steps drawn from a small fixed pool, including “Let me restate what we have so far: [paraphrase].”, “To summarise the progress so far, [paraphrase].”, and “Before continuing, I’ll verify the current state: [paraphrase].” Confidence-injection attacks prepend each step with a phrase drawn uniformly from the set {“By a well-known

Table 12: Concrete illustrations of the three EST-PRM attacks on a short example chain. In every case the final answer ($\boxed{10}$) is preserved. The position attack uses a reward-front-loaded topological ordering subject to dependency constraints, with the final-answer step fixed.

Attack	Example transformation (final answer preserved)
Honest chain	s_1 : First, $a = 2 \cdot 3 = 6$. s_2 : Independently, $b = 5 - 1 = 4$. s_3 : Then $a + b = 10$. $\boxed{10}$
Step-inflation	s_1 : First, $a = 2 \cdot 3 = 6$. <i>[Let me restate: a is now defined.]</i> s_2 : Independently, $b = 5 - 1 = 4$. <i>[To summarise: both a and b are computed.]</i> s_3 : Then $a + b = 10$. $\boxed{10}$
Position-sensitivity	s_2 : Independently, $b = 5 - 1 = 4$. s_1 : First, $a = 2 \cdot 3 = 6$. s_3 : Then $a + b = 10$. $\boxed{10}$
Confidence-injection	s_1 : Clearly, $a = 2 \cdot 3 = 6$. s_2 : By definition, $b = 5 - 1 = 4$. s_3 : It is immediate that $a + b = 10$. $\boxed{10}$

result,” “It is immediate that,” “Clearly,” “Without loss of generality,” “By definition,” “Trivially,” “Obviously,”}.

Position-sensitivity attacks use a single pre-specified reward-front-loaded topological ordering. We first construct a dependency graph over reasoning steps and permit a step to move earlier only when the resulting ordering preserves all required dependencies. The dependency graph includes variable-definition and consumption edges, equation–result dependencies, explicit discourse references, coreference relations, case and scope dependencies, and local justification dependencies. The final-answer step is always fixed at its original position. Among otherwise eligible steps, ties are broken by the original step index. The resulting ordering therefore front-loads the highest-PRM-reward eligible steps subject to the structural constraints, without altering the final answer or violating required reasoning dependencies.

To validate the position construction, we conducted a blind audit of 300 position-transformed chains. Of these, 239 satisfied the strict eligibility criteria. Across the full evaluation set, 3,742 chains were strict-eligible under the same procedure, with a residual invalidity rate of 2.5%. These stricter dependency checks ensure that the position transformation changes the ordering of eligible reasoning steps without introducing new mathematical or semantic errors.

H Chain-Level Aggregation Sensitivity

We evaluate five deterministic chain-level aggregators using the same native per-step scores: arithmetic mean, minimum, maximum, final-step score, and geometric mean. For position sensitivity, all aggregators use the same fixed reward-front-loaded order; no row takes a maximum over

multiple position strategies.

Table 13: Correlation collapse under different chain-level aggregation rules.

Model	Aggregation	Step	Position	Confidence
Math-Shepherd	Mean	0.128	0.139	0.011
	Minimum	0.070	0.092	0.008
	Maximum	0.105	0.084	0.013
	Final step	0.046	0.089	0.009
	Geometric mean	0.119	0.141	0.011
Qwen2.5-PRM	Mean	0.089	0.035	0.015
	Minimum	0.053	0.028	0.011
	Maximum	0.076	0.031	0.017
	Final step	0.036	0.020	0.010
	Geometric mean	0.083	0.033	0.014

Table 13 shows that structural sensitivity persists across the tested aggregation rules, while its magnitude and, in some cases, dominant attack depend on the aggregation rule. Across the ten displayed model–aggregation settings, the dominant attack profile is unchanged in nine; the exception is Math-Shepherd under maximum aggregation, where the dominant sensitivity shifts from position to step-inflation. We therefore treat aggregation as a material design choice rather than assuming that the arithmetic mean is uniquely representative.

I Additional Discrimination Diagnostics

To further characterize the baseline discrimination of Math-Shepherd-7B, we report AU-ROC, AUPRC, and balanced accuracy alongside the point-biserial Pearson correlation for the composite-position analysis. The baseline PRM provides nontrivial discrimination, while the composite-position transformation substantially reduces this signal. Table 14 reports the corresponding discrimination metrics for the honest and attacked chains.

Table 14: Discrimination diagnostics for Math-Shepherd-7B under the composite-position transformation.

	Honest	Attacked	Change
Pearson/point-biserial	0.381 ± 0.012	0.229 ± 0.034	-0.152 ± 0.038
AUROC	0.741 ± 0.012	0.653 ± 0.015	-0.088 ± 0.014
AUPRC	0.736 ± 0.013	0.646 ± 0.016	-0.090 ± 0.016
Balanced accuracy	0.687 ± 0.013	0.612 ± 0.014	-0.075 ± 0.015

J Naturalistic Structural Edits

We conduct a template-free naturalistic-edit experiment on 600 held-out chains using Meta-Llama-3-70B-Instruct with the instruction: “make the solution clearer, more explicit, and easier to follow without changing any mathematical claim or the final answer.” The prompt contains no EST-PRM attack instructions. Three blinded annotators validate the edits using the same preservation rubric; 552/600 (92.0%) pass. Among valid edits, 58.2% contain redundant restatement, 24.6% confidence/discourse markers, and 12.9% local reordering. The controls use the same 552 source chains, scoring interface, aggregation rule, and problem-cluster bootstrap as the naturalistic-edit evaluation. One matched paraphrase control is generated for each valid naturalistic edit, and all 552 controls pass the preservation check. Naturalistic edits therefore have the same denominators as the controls and produce larger correlation collapse, inflation rates, and mean score changes. The effects are smaller than those of the targeted EST-PRM transformations but retain the same direction and model-specific pattern. Table 15 summarizes the results for the 552 validated naturalistic edits and their matched paraphrase controls.

K Dense Stepwise LLM-Critic Control

We additionally evaluate a dense stepwise Meta-Llama-3-8B-Instruct critic, assigning one correctness probability per step and mean-aggregating them, with the same position transformation. Table 16 shows substantially smaller correlation collapse than the most vulnerable PRMs, supporting the scalar critic as a control comparison rather than an aggregation artifact.

L Dependency construction and rejection categories

For each reasoning chain, we construct a dependency graph whose nodes are reasoning steps and whose directed edges encode constraints that must

be preserved under reordering. Edges include variable-definition and consumption dependencies, equation–result dependencies, explicit discourse and local-justification dependencies, coreference and ellipsis relations, and assumption or case-scope dependencies. Candidate position transformations are generated only from topological orders of this graph. The final-answer step is fixed at its original position, and among otherwise eligible steps the reward-front-loaded order is selected using the native PRM score, with the original step index as the tie-breaker.

The complete position-eligibility procedure is summarized in Algorithm 1. In brief, the procedure constructs the dependency graph, restricts candidates to valid topological orders, applies explicit discourse, coreference, scope, and local-justification checks, fixes the final-answer step, and selects the reward-front-loaded order among the remaining eligible steps.

For transparency, rejected position candidates are assigned a primary failure category. Table 17 reports the distribution of these rejection categories among rejected candidates. These categories describe why a candidate transformation is ineligible; they are not interpreted as evidence that every rejected candidate contains a mathematical error.

All attempted transformations, including accepted and rejected candidates, are recorded in the transformation ledger together with their eligibility flags and rejection reasons. The automated eligibility procedure is complemented by a stratified blind human audit used to estimate residual invalidity; the audit is not treated as a per-chain human filtering step.

M Scoring, Inference, Calibration, and Implementation Details

The main chain score is the mean native step score. Minimum, maximum, and final-step scores are direct deterministic sensitivity analyses. For

Table 15: PRM behavior on 552 validated naturalistic edits and paraphrase controls.

	Math-Shepherd		Qwen2.5-Math-PRM	
	Naturalistic	Control	Naturalistic	Control
Correlation collapse ($\Delta\rho$)	0.067 $\pm$ 0.018	0.009 $\pm$ 0.004	0.061 $\pm$ 0.016	0.012 $\pm$ 0.005
Inflation rate	16.9 $\pm$ 2.6%	4.8 $\pm$ 1.0%	27.8 $\pm$ 3.4%	6.3 $\pm$ 1.4%
Mean score change	+0.010 $\pm$ 0.004	+0.001 $\pm$ 0.002	+0.019 $\pm$ 0.005	+0.003 $\pm$ 0.002

Table 16: Dense stepwise critic: correlation collapse.

Configuration	Step	Position	Confidence
Mean	0.018 $\pm$.006	0.022 $\pm$.007	0.009 $\pm$.003
Minimum	0.013 $\pm$.005	0.017 $\pm$.006	0.008 $\pm$.003

Algorithm 1 Strict position-eligibility and ordering

Require: Reasoning chain $x = (s_1, \dots, s_k)$ and native step scores $r_1, \dots, r_k$

Ensure: Eligible transformed view $A(x)$ or rejection

- 1: Construct dependency graph $G = (V, E)$ over the reasoning steps
- 2: Add definition/use, equation–result, discourse, coreference, scope, and local-justification edges
- 3: Enumerate only orders consistent with G
- 4: Fix the final-answer step at its original position
- 5: Exclude orders violating discourse, coreference, scope, or local-justification constraints
- 6: Among remaining eligible steps, sort by decreasing r_i with original index as the tie-breaker
- 7: **if** no valid order exists **then**
- 8: **return** Reject with eligibility reason
- 9: **else**
- 10: **return** the resulting reward-front-loaded order $A(x)$
- 11: **end if**

probability-valued native scores, geometric mean is $\exp(k^{-1} \sum_i \log(\max(r_i, 10^{-8})))$. The inflation statistic is

$$I_{0.1} = \frac{1}{n} \sum_j \mathbf{1}[S(Ax_j) > 1.1S(x_j)],$$

where $S(x_j)$ and $S(Ax_j)$ denote the native and transformed chain scores, respectively. Paired mean score change is reported separately. with paired mean score change reported separately. Submitted results retain their reported uncer-

tainty intervals; all new strict-subset, naturalistic-edit, dense-critic, discrimination, selection, aggregation, and mitigation analyses use 10,000 problem-cluster bootstrap replicates, keeping all chains/views from a problem within each resample.

Balanced-accuracy thresholds are selected within each training fold to maximize balanced accuracy on honest/corrupted training chains, with ties broken by the largest threshold, and evaluated once on held-out problems. Mitigation uses five-fold problem-level cross-fitting: normalization and the OLS score-versus-step-count residual are fitted on 800 problems, the smallest threshold with empirical honest-chain FPR $\leq 5\%$ is selected, and evaluated on 200 held-out problems; no attacked view is used for fitting.

The dense critic uses Meta-Llama-3-8B-Instruct with deterministic decoding. The prompt is: “For each numbered reasoning step, return only JSON of the form `{"step_probabilities": [p1, ..., pk]}`, where p_i is the probability that step i is mathematically correct given the question and all preceding steps.” Outputs are checked for JSON syntax, list length, numeric type, and probability range, with no repair prompt. Invalid outputs are not silently dropped. Parser-valid/attempted counts are 4,687/4,687 for the original, step-inflation, and confidence views, and 3,742/3,742 for fixed position; metrics use these valid intersections. Each run records the resolved repository revision SHA, native chat/reward template, maximum input length, truncation side, and chains excluded for exceeding the native limit. The per-problem ledger records source-chain identifiers and accepted/rejected transformation flags, while problem-cluster resampling ensures

Table 17: Rejection categories among position candidates. Percentages are computed over rejected candidates.

Category	Share	Example
Definition/use violation	34.2%	Moving a step that uses x before the step defining x
Equation–result dependence	22.8%	Moving substitution before the equation whose solution is used
Discourse/local justification	18.4%	Moving “therefore” away from the premise it summarizes
Assumption/case-scope violation	13.2%	Moving a case-specific conclusion outside its case
Coreference/ellipsis ambiguity	7.9%	Moving “this value” before its antecedent
Other/annotator disagreement	3.5%	Grammatically valid but locally ambiguous presentation

that chains from the same problem are never treated as independent.

N Reproducibility Protocol

We provide the implementation details used for the revised analyses. The protocol is summarized in Table 18, including dataset composition, chain generation, step segmentation, correctness labeling, attack construction, paired controls, eligibility criteria, and handling of missing views. The table specifies the prompts, filtering rules, seeds, and other analysis settings used to obtain the reported results.

O Supplementary Figures

This section provides additional visual analyses supporting the EST-PRM evaluation. The figures summarize the evaluation design, attack effects, influence rates, model-specific robustness patterns, mitigation behavior, reproducibility components, and dataset filtering statistics.

Figure 4 provides a high-level comparison between EST-PRM and prior reward-model evaluation benchmarks, emphasizing the role of PRM-specific and dense adversarial evaluation.

Figure 5 further examines how different attack types affect model scores and influence estimates for Math-Shepherd-7B. As shown in Figure 5, Step-inflation and Position Sensitivity produce substantially larger effects than Confidence perturbations, with Position Sensitivity exhibiting the highest $\Delta\rho$ and Influence Rate, while Confidence remains consistently close to zero across all measures.

Figure 6 visualizes the attack-induced correlation changes across the six evaluated models. The results reveal substantial variation in attack sensitivity: Step-inflation produces the largest $\Delta\rho$ for Qwen2.5-Math-PRM-7B and DeepSeek-Math-PRM, whereas Position is dominant for Math-Shepherd-7B and Skywork-o1-Open-PRM. In contrast, RLHFlow-PRM-8B ex-

PRM Evaluation Methodology Comparison

Method	PRM-specific (Targets PRM weaknesses directly)	Dense attack (Evaluates dense-reward hacking)
ProcessBench	✓	—
PRMBench	✓	—
RewardBench	—	—
RLHF Hacking	—	—
EST-PRM (ours)	✓	✓

✓ Yes (Capability Present)
— No (Capability Absent)

Figure 4: Comparison of PRM evaluation methodologies across existing benchmarks and the proposed EST-PRM framework. The figure illustrates whether each method is PRM-specific and whether it incorporates dense adversarial attacks. Prior work, including ProcessBench, PRMBench, RewardBench, and RLHF Hacking, primarily evaluates models under natural or outcome-level errors with limited consideration of dense adversarial settings. In contrast, EST-PRM explicitly integrates both PRM-specific evaluation and dense adversarial perturbations (Step-inflation, Position, and Confidence attacks), enabling a more comprehensive assessment of reward model robustness under realistic and adversarial conditions.

Table 18: Complete reproducibility specification for the experimental protocol.

Component	Exact specification for analyses
Dataset composition	1,000 problems: 400 MATH-500, 400 GSM8K, and 200 PRMBench; five generations per problem before preprocessing; 4,687 retained source chains.
Chain generation	Meta-Llama-3-70B-Instruct; temperature 0.7; one generation call per candidate; global seed 42; prompt requires separately delimited reasoning steps and a final boxed answer.
Step segmentation	Split first on explicit line breaks. Split a line further only at ., ?, or ! followed by whitespace and an uppercase letter, digit, or opening parenthesis; exclude decimal points, common abbreviations, and text inside LaTeX math spans. Remove empty segments and attach a standalone boxed-answer segment to the preceding step.
Correctness labels	Extract the final boxed answer; test algebraic equivalence with SymPy; manually adjudicate extraction failures; assign the source-chain binary label before transformation.
Step-inflation count and locations	For k reasoning steps, insert $m = \min(4, k - 1)$ restatements at deterministic equally spaced quantiles of the $k - 1$ non-final boundaries. The retained segmented pool contains no $k = 1$ chain.
Step-inflation payload	Meta-Llama-3-70B-Instruct with deterministic decoding. Exact prompt: Restate only the mathematical information already established before this boundary in one sentence. Do not introduce a new variable, number, premise, operation, or conclusion. Return only the restatement. The returned sentence is wrapped in one of three fixed Appendix-F templates selected by seed 42. Each boundary receives one deterministic generation attempt. If any payload fails the stated validity checks, the entire paired attack/control view is excluded; there is no regeneration. The saved ledger records 4,687 accepted source-chain pairs and 0 rejected pairs.
Confidence injection	Prefix each reasoning step, excluding a standalone boxed-answer token, with one marker sampled independently and uniformly from the seven fixed Appendix-F markers using seed 42.
Revised primary position rule	One reward-front-loaded topological order per strict-eligible chain. Among available nodes, choose the node with the largest original native step reward; break ties by original step index; fix the final-answer step. Reverse- topological and reward-back-loaded orders are separate sensitivity rows.
Submitted position result	Retained as the originally reported composite result because the manuscript did not specify its three-strategy combination clearly enough. It is not used as the revised primary validity or downstream result.
Position eligibility	Add variable definition/use, equation-result, explicit-reference, case/scope, final-answer, and local-justification edges. Automatic discourse checks use the fixed marker set {therefore, hence, thus, consequently, from the previous step, using the above, this value, this result, the former, the latter} plus antecedent availability. Unresolved pronoun/noun-phrase antecedents or case-scope markers cause rejection.
Exact paired multiplicity controls	Position: a multiplicity-matched formatting view using numbering, whitespace, and line-wrapping changes only; it is not called a semantically matched reorder. Step inflation: the same count and boundary locations with fixed neutral strings {Continuing the derivation., Proceeding to the next step., We now continue.} cycled by seed 42. Confidence: position-matched neutral prefixes {For this step,, At this point,, In the next calculation,}. An attack view is included only if its paired control is also valid.
Naturalistic edit prompt	Meta-Llama-3-70B-Instruct with the chain-generation decoding configuration. Exact instruction: Make the solution clearer, more explicit, and easier to follow without changing any mathematical claim or the final answer. The prompt contains no EST-PRM attack name, template, confidence phrase, or reordering instruction.
Matched-paraphrase prompt	Paraphrase each numbered step independently. Preserve the number and order of steps, every equation, every numerical value, all assumptions, and the final answer. Do not add or remove reasoning. Failed/non-preserving outputs are excluded. Identical outputs are retained and counted as zero-change controls.
View-consistency missing views	If the paraphrase fails, use the original and formatting views and record a missing-paraphrase indicator; do not regenerate. An identical paraphrase remains a valid zero-variation view.

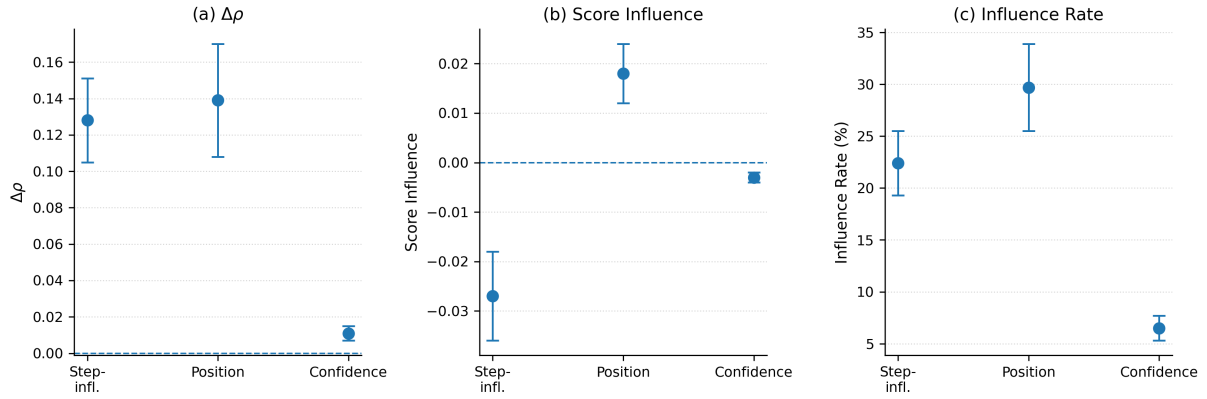


Figure 5: Comparison of attack effects on the Math-Shepherd-7B model under Step-inflation, Position Sensitivity, and Confidence perturbations. The figure reports $\Delta\rho$, Score Influence, and Influence Rate with corresponding uncertainty intervals. Step-inflation and Position Sensitivity produce substantially larger effects than Confidence perturbations. Position Sensitivity yields the largest $\Delta\rho$ and Influence Rate, together with a positive Score Influence, whereas Confidence perturbations remain close to zero across all three measures.

hibits a negative response to Step-inflation, while the Llama-3-8B-Critic remains comparatively insensitive across all three attacks.

Figure 7 further examines model-specific exploitation risk by comparing the proportion of evaluations exhibiting a relative score increase greater than 10%. As shown in Figure 7, Qwen2.5-PRM exhibits the highest vulnerability to Step-inflation, with an inflation rate of 47.6%, while DeepSeek-PRM shows a similarly elevated rate of 38.8%. Position-based attacks are particularly effective against Math-Shepherd and Skywork-PRM, reaching 29.7% and 30.2%, respectively. In contrast, RLHFlow-PRM remains comparatively resistant across all three attack types.

Figure 8 visualizes the change in sample counts before and after filtering invalid or non-parsable chains. As shown in Figure 8, all three datasets retain the majority of their initial chains, with MATH-500 decreasing from 2,000 to 1,812, GSM8K from 2,000 to 1,930, and PRMBench from 1,000 to 945. Overall, 4,687 of the initial 5,000 chains are retained for the main experiments.

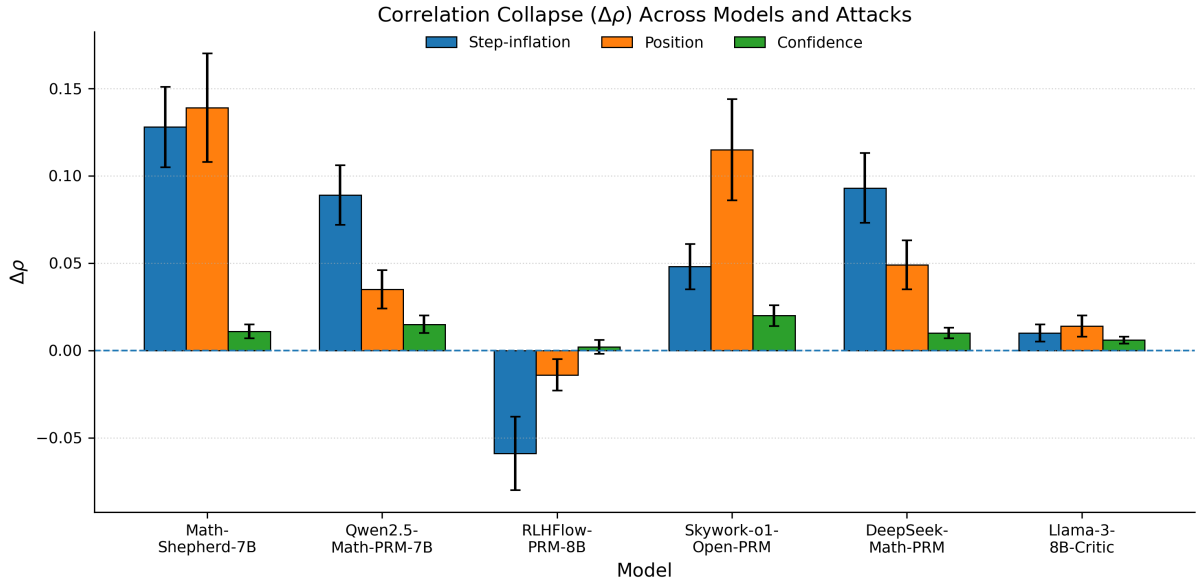


Figure 6: Correlation collapse ($\Delta\rho$) across six process-reward-style models and three attacks. Bars show mean $\Delta\rho$ with 95% bootstrap confidence intervals. Step-inflation produces the largest collapse for Qwen2.5-Math-PRM-7B and DeepSeek-Math-PRM, whereas Position is dominant for Math-Shepherd-7B and Skywork-o1-Open-PRM. RLHFlow-PRM-8B exhibits a negative response to Step-inflation, while the Llama-3-8B-Critic remains comparatively insensitive across all attacks.

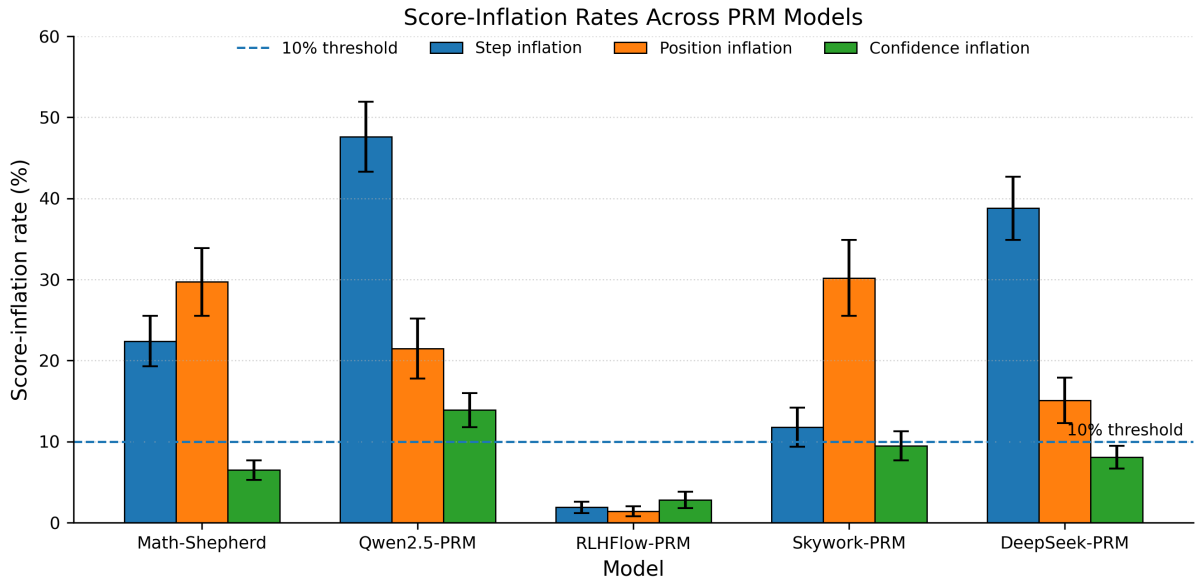


Figure 7: Score-inflation rates for five process-reward models under Step-inflation, Position inflation, and Confidence inflation attacks. Bars show the percentage of evaluations exhibiting a relative score increase greater than 10%, with uncertainty bars representing the reported variability. The dashed line marks the 10% exploitation threshold. Qwen2.5-PRM and DeepSeek-PRM exhibit particularly high vulnerability to Step-inflation, whereas Position attacks are most effective against Math-Shepherd and Skywork-PRM.

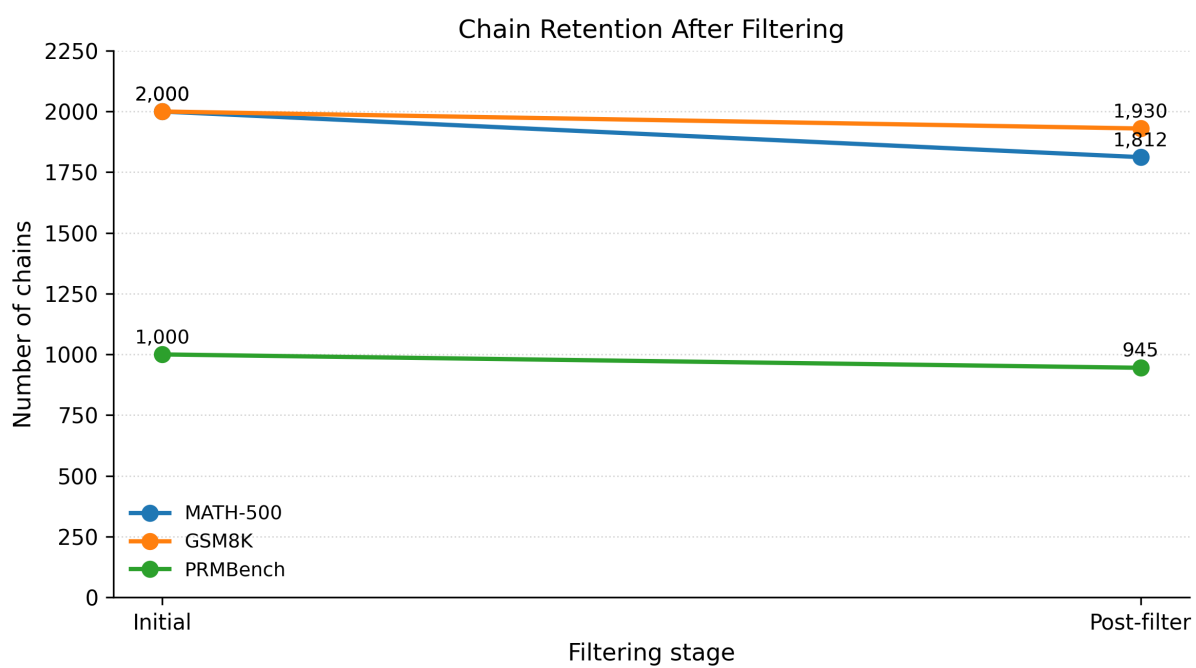


Figure 8: Sample counts before and after filtering invalid or non-parsable chains. The graph shows the retained chains for MATH-500, GSM8K, and PRMBench, with the overall pool decreasing from 5,000 initial chains to 4,687 retained chains used in the main experiments.