THE SOLUTION IS THE SUPERVISION: LEARNING TO VERIFY REASONING WITHOUT HUMAN STEP LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome reward models assess final-answer correctness in mathematical reasoning, while process reward models (PRMs) assess intermediate steps. Implicit PRMs learn local scores from outcome labels, avoiding costly human step annotations. But fitting a response's outcome does not directly supervise where its derivation first becomes invalid. We investigate this distinction between outcome fitting and local judgment, and ask what worked reference solutions add to verification supervision. An aggregate-score analysis and a trained-model diagnostic show that small outcome residuals can coexist with unresolved first-error judgments. To study reference information, we use on-policy self-distillation to transfer a same-backbone frozen teacher's reference-conditioned evaluations to a reference-free student. On ProcessBench, complete references improve macro F1 by 3.58 points over answer-only supervision and exact localization by 5.57 points on correct-answer solutions with process errors. Content and alignment interventions connect these improvements to information in the worked derivation. A separate candidate-composition study finds a larger advantage over Implicit CE when successful candidates are scarce, with the endpoint effect reproduced in a second training seed. Together, the results characterize a local-supervision gap and show how reference-informed evaluation can improve judgments without human candidate-step labels.

1 INTRODUCTION

In mathematical reasoning with large language models (LLMs), reward models help select candidate solutions and guide model training. Outcome reward models (ORMs) learn from final-answer correctness, whereas process reward models (PRMs) provide feedback on intermediate steps (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024). Intermediate judgments matter when an apparently successful answer rests on invalid reasoning. In the schematic example of Figure 1, the candidate writes $8/2 = 5$ but ends with the correct answer $x = 4$. An outcome label records success without identifying the incorrect calculation.

The difficulty is obtaining supervision for these judgments. Human step annotations are expensive, and automatic methods construct labels from sampled continuations, search, or a prover's progress (Wang et al., 2024; Luo et al., 2024; Setlur et al., 2025). Implicit PRM provides another route: outcome training can recover useful local rewards through token log-likelihood ratios (Yuan et al., 2025). This avoids collecting step labels while retaining local scores for downstream use. Its success raises a further question: how well do those scores identify logically invalid steps when the final outcome is an incomplete guide to the process?

Our starting point is the distinction between computing a local reward and supervising a local judgment. Outcome fitting need not resolve the first-error decision. We examine this distinction through the aggregate objective, a trained-model diagnostic, and solutions whose correct answers conceal process errors (Zhang et al., 2025b). We also vary the proportion of successful training candidates to study how the two supervision routes respond to candidate composition.

Complete worked solutions provide intermediate relationships as well as a final answer. They can support checking the candidate's calculation in Figure 1, even though its outcome label is positive.

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

The idea resembles a teacher consulting a worked solution while guiding an apprentice grader: the apprentice cannot see the reference and learns to assess the work independently. This use of privileged training information (Vapnik & Izmailov, 2015) requires both a reference-informed evaluator and a way to transfer its behavior to a student given only the problem and candidate.

We transfer reference-informed judgments using on-policy self-distillation (OPSD) (Zhao et al., 2026b). A frozen same-backbone teacher reads the reference and supplies distributions at the student's own evaluation prefixes. With the candidate fixed, only the student adapter is trained. The resulting OPSD-PRM verifies without references; both successful and failed candidates supply evaluation targets without discrete candidate-step labels.

We make three contributions. First, we connect aggregate-supervision analysis to a trained-model diagnostic of small outcome residuals coexisting with localization errors (Section 2). Second, reference-content and alignment interventions show that worked derivations improve independent verification, including localization of errors concealed by correct answers (Section 5). Third, OPSD's relative advantage over Implicit CE increases at low success proportions, with endpoint replication in two training seeds (Section 5.5). Section 6 examines training budgets and alternative targets.

2 Outcome Fitting and First-Error Judgments

We connect the constraints imposed by outcome supervision to a trained-model diagnostic of outcome fitting and first-error judgment.

A response score and its local terms. Let $y = (y_1, \ldots, y_T)$ be a response to problem x, B_ϕ a trainable model, and C a fixed reference model. Implicit PRM parameterizes a response score as a sum of token log-likelihood ratios (Yuan et al., 2025):

$$R_\phi(x, y) = \sum_{t=1}^{T} r_t, \qquad r_t = \beta \log \frac{B_\phi(y_t \mid x, y_{<t})}{C(y_t \mid x, y_{<t})}. \tag{1}$$

For a loss depending on ϕ only through these response scores, differentiation with responses held fixed gives

$$\nabla_\phi \mathcal{L} = \beta \sum_i \underbrace{\frac{\partial \mathcal{L}}{\partial R_\phi(x_i, y_i)}}_{\text{shared within response } i} \sum_t \nabla_\phi \log B_\phi(y_{i,t} \mid x_i, y_{i,<t}). \tag{2}$$

Equation 2 separates the shared response-level coefficient from context-dependent token gradients. For binary cross-entropy (CE), this coefficient is $\sigma(R_\phi) - l$, where l is the final-answer correctness label.

Same total, different local judgments. Group the token rewards in Equation 1 into step scores $\mathbf{s} = (s_1, \ldots, s_K)$. Treating these scores as free coordinates, a single-candidate loss $\ell(\sum_k s_k, l)$ is unchanged by any perturbation $(\delta_1, \ldots, \delta_K)$ satisfying $\sum_k \delta_k = 0$. For example,

step scores	$\longrightarrow$	cumulative scores
$(2, -3, 4)$	$\longrightarrow$	$(2, -1, 3)$
$(1, 1, 1)$	$\longrightarrow$	$(1, 2, 3)$

Both allocations have total score 3 and the same outcome loss. With an illustrative zero threshold on cumulative scores, the first flags the second step; the second accepts all steps. Thus, the aggregate loss alone does not specify the local allocation in this score space. In a language model, shared parameters, probability normalization, and other examples constrain which allocations are realizable. Appendix A.2 gives the corresponding curvature analysis.

Small residual, unresolved localization. The CE coefficient can be small without checking the first-error decision. We examine this directly using the GSM8K-selected CE checkpoint from the 1,024-candidate study (Section 6.1). Define the outcome residual as $e = |\sigma(R_\phi) - l|$ and retain the fixed GSM8K-selected localization threshold. The three test subsets, MATH, OlympiadBench,

and Omni-MATH (Test-3), contain 1,500 candidates with incorrect final answers and process errors; 924 satisfy $e \leq 0.10$. Of these, 698 still have incorrect first-error predictions: 75.54%, with a 95% problem-cluster bootstrap interval of $[72.65, 78.30]$. Small outcome residuals and localization errors therefore coexist at this checkpoint. Only one correct-answer candidate with process errors meets the residual condition, so this diagnostic concerns failed answers. Appendix A.4 reports the full protocol and comparisons on the CE-conditioned subset.

From diagnosis to reference supervision. A failed answer may contain valid steps, while the correct answer in Figure 1 conceals an invalid calculation. Prefix value for eventual success and current-step validity are distinct targets (Yuan et al., 2025; Setlur et al., 2025). Worked references provide intermediate relationships for checking candidates. We transfer this information through teacher targets for student evaluation text. The residual diagnostic characterizes outcome fitting; separate reference interventions test changes in student judgments, and a fixed-problem composition study examines scarcity of successful candidates.

3 Reference-Guided Evaluation through Self-Distillation

Figure 1 shows how privileged teacher inputs train a verifier with a reference-free student interface.

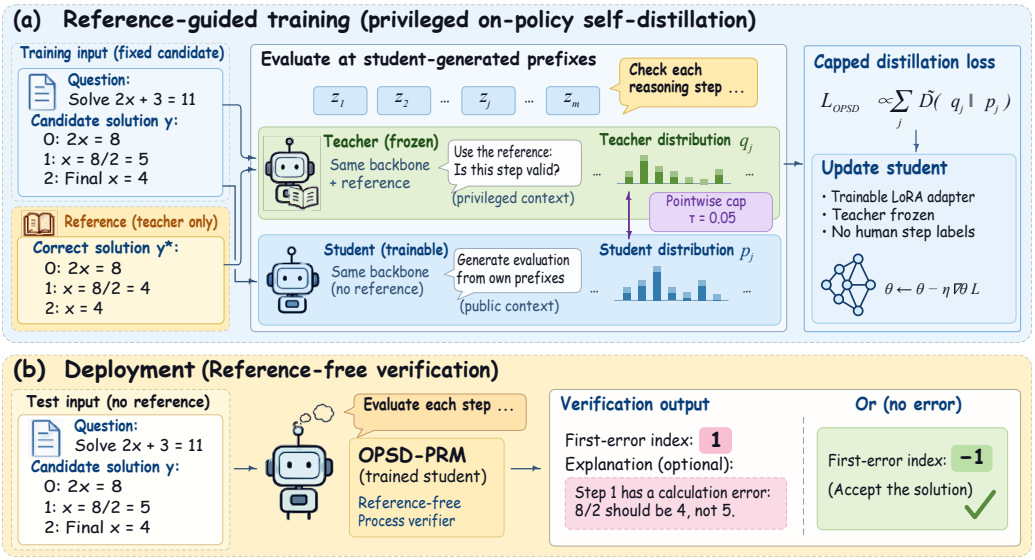


Figure 1: Reference information guides training; deployment uses the student alone. (a) For a fixed candidate, the reference-conditioned frozen teacher supervises student-generated evaluation prefixes; only the student adapter is updated. (b) The student returns the first-error index or -1 without references. The example and distributions are schematic; Equation 4 specifies the capped objective.

3.1 Evaluating a Fixed Candidate with a Worked Reference

For problem x and candidate steps $y = (s_0, \ldots, s_{K-1})$, let $u = (x, y)$ denote the public evaluation input, including the task instruction. A worked reference r is available only during training. We instantiate the frozen teacher from Qwen2.5-7B-Instruct and initialize the student from the same backbone (Qwen Team, 2024), updating only its LoRA adapter (Hu et al., 2022). The teacher receives the original reference, including its derivation and any final answer appearing in it; the student receives the complete candidate but no reference.

In Figure 1, the reference supports checking $8/2$ despite the correct final answer. The teacher supervises evaluation text without constructing discrete candidate-step labels.

3.2 Distillation at Student Evaluation Prefixes

To guide evaluations at states visited by the student, we use OPSD (Zhao et al., 2026b) with a frozen initial-model teacher. The student samples an evaluation $z \sim p_\theta(\cdot \mid u)$ for the fixed candidate. Both contexts receive the same evaluation-prefix token IDs:

$$q_j(v) = \mathrm{sg}\; p_{\theta_0}^{(T)}(v \mid u, r, z_{<j}), \qquad p_j(v) = p_\theta^{(T)}(v \mid u, z_{<j}), \quad v \in \mathcal{V}. \tag{3}$$

In Equation 3, θ_0 is the frozen initial model, sg stops gradients, and $T = 1.1$ is the distribution temperature. For each sampled evaluation, we optimize OPSD's signed pointwise-capped forward-KL objective:

$$\ell_\theta(u, r, z) = \frac{1}{|z|} \sum_{j=1}^{|z|} \sum_{v \in \mathcal{V}} \min \left\{ q_j(v) \log \frac{q_j(v)}{p_j(v)}, \tau \right\}, \qquad \tau = 0.05. \tag{4}$$

The batch objective averages Equation 4 over examples. The pointwise cap follows OPSD: it is applied before vocabulary summation and retains negative contributions. Sampled tokens and teacher targets are fixed within an update; gradients pass only through student probabilities. Prompt and padding positions are excluded. The updated student generates new evaluations for subsequent training examples.

The teacher provides evaluation targets for both successful and failed candidates. Appendices B and A give the implementation and capped gradient; Section 6.2 evaluates the cap on this task.

3.3 Reference-Free First-Error Verification

The student reads the problem and numbered candidate paragraphs, generates an evaluation, and returns a boxed zero-based first-error index $\hat{g} \in \{0, \ldots, K-1\}$, or -1 for a fully correct solution. It judges reasoning validity even when the final answer is correct, with the complete candidate available for this retrospective judgment. Evaluation uses one greedy continuation of at most 1,024 tokens; invalid and out-of-range verdicts count as failures.

4 Experimental Setup

Data and training. We use problems, pre-generated candidates, and native worked references from the PRM800K phase-2 training release (Lightman et al., 2024). Human step ratings and verifier scores are excluded from filtering and training targets. The main model and reference interventions use 6,400 distinct candidates from 4,247 problems, one visit each, and 200 updates. All use Qwen2.5-7B-Instruct, LoRA rank 64, global batch 32, and learning rate 5×10^{-6} with linear decay. The normalized exact problem intersection with ProcessBench is zero. Appendix B.1 specifies the source and preparation.

Matched comparisons. The budget study uses nested pools of 256–2,048 problems, one candidate each, visited twice. The 1,024-candidate pool also supports learning-rate selection, teacher-evaluation supervised fine-tuning (SFT), and uncapped-KL controls, all at 64 updates. Implicit CE fits outcome labels; Math-Shepherd-style Monte Carlo hard estimation (MC-HE) labels each non-terminal prefix from eight continuations. OPSD generates evaluations after reading complete candidates; the score models read causal prefixes. Reference interventions keep the student interface fixed. A separate paired-candidate pool supports the composition intervention. Published rows in Table 1 retain their source protocols; local EurusPRM-Stage1 predictions supply the paired external comparison (PRIME-RL, 2025a).

Metrics and calibration. ProcessBench contains 3,400 candidates across GSM8K, MATH, OlympiadBench, and Omni-MATH (Zheng et al., 2025). For each subset, $F1 = 2EC/(E + C)$, where E is exact first-error accuracy on erroneous candidates and C is acceptance of fully correct candidates. Invalid outputs remain failures in the denominators. Macro-4 averages all four subset F1 values. Score models select a threshold on GSM8K's 400 examples over $i/1999$, $i = 0, \ldots, 1999$, with smallest-threshold tie breaking. Test-3 averages the remaining subsets. The learning-rate check also selects on GSM8K; generated verdicts use no threshold.

Table 1: ProcessBench F1 (%); full model list in Appendix D.4. Bold marks block maxima. †: Zheng et al. (2025, Table 3); ‡: Zhao et al. (2026a), v1, Table 1. Published protocols are retained: critics use eight-sample voting; OPSD uses greedy decoding. Macro-4 averages the subsets.

Model	GSM8K	MATH	Olympiad Bench	Omni MATH	Macro-4
Prompted language-model critics (7–8B)					
Qwen2.5-Math-7B-Instruct†	26.8	25.7	14.2	12.7	19.9
Qwen2.5-7B-Instruct†	**36.5**	**36.6**	**29.7**	**27.4**	**32.6**
Other process reward models					
Qwen2.5-Math-7B-PRM800K†	68.2	**62.6**	**50.7**	**44.3**	**56.5**
Skywork-7B†	**70.8**	53.6	22.9	21.0	42.1
EurusPRM-Stage2‡	47.3	35.7	21.2	20.9	31.3
Math-Shepherd-7B†	47.9	29.5	24.8	23.8	31.5
Implicit baseline and reference-distilled verifier					
EurusPRM-Stage1 (reported)‡	44.3	35.6	21.7	23.1	31.2
OPSD-PRM (ours)	**52.94**	**43.28**	**30.93**	**34.39**	**40.38**

Uncertainty. The main model has three training seeds; the reference interventions use seed 20260914. The composition endpoints have seeds 20260914 and 20260915 with candidate identities and visit orders fixed. Other configurations have one seed. Paired 95% intervals use 5,000 problem-cluster bootstrap draws within subsets, seed 42, and fixed models and thresholds. They describe evaluation-sample uncertainty; results for different training seeds are reported separately.

5 Reference Information and Local Verification

We report overall performance, reference interventions, and local judgments, followed by the fixed-problem candidate-composition study.

5.1 Overall Process-Verification Performance

OPSD-PRM achieves 40.38 Macro-4 F1 (Table 1), with 40.50 ± 0.76 across three training seeds. Under the same public prompt and greedy readout, the untrained backbone scores 27.82 (Appendix E). OlympiadBench and Omni-MATH improve from 18.57 and 19.75 to 30.93 and 34.39. The table gives EurusPRM-Stage1's published 31.2; the local run used for paired diagnostics scores 34.52. Appendix D.4 records the sources and supervision resources.

5.2 Reference Information in Teacher Judgments

The frozen base model evaluates 512 candidates with no reference, a matched full reference, a shuffled reference, or only the final answer. All four conditions continue from identical evaluation-prefix token IDs sampled once by the reference-free base model. Reference access raises Macro-4 F1 from 32.10 to 38.71, a gain of 6.61 points $[2.41, 10.83]$. Matched references also improve over shuffled references by 7.78 points $[3.58, 12.07]$ (Figure 2a). The additional gain over answer-only is 3.53 points $[-0.26, 7.40]$. Appendix C gives all four conditions and the 63-example correct-answer error subset.

5.3 Learning from Reference Content and Alignment

The student interventions change only the teacher's reference content, retaining the 200-update training configuration and reference-free student interface. Answer-only removes the derivation; explicit-answer masking removes boxed/fboxed spans; shuffling assigns a different problem's reference while preserving the reference-text multiset. Masking can leave answer information elsewhere in the reference (Appendix C).

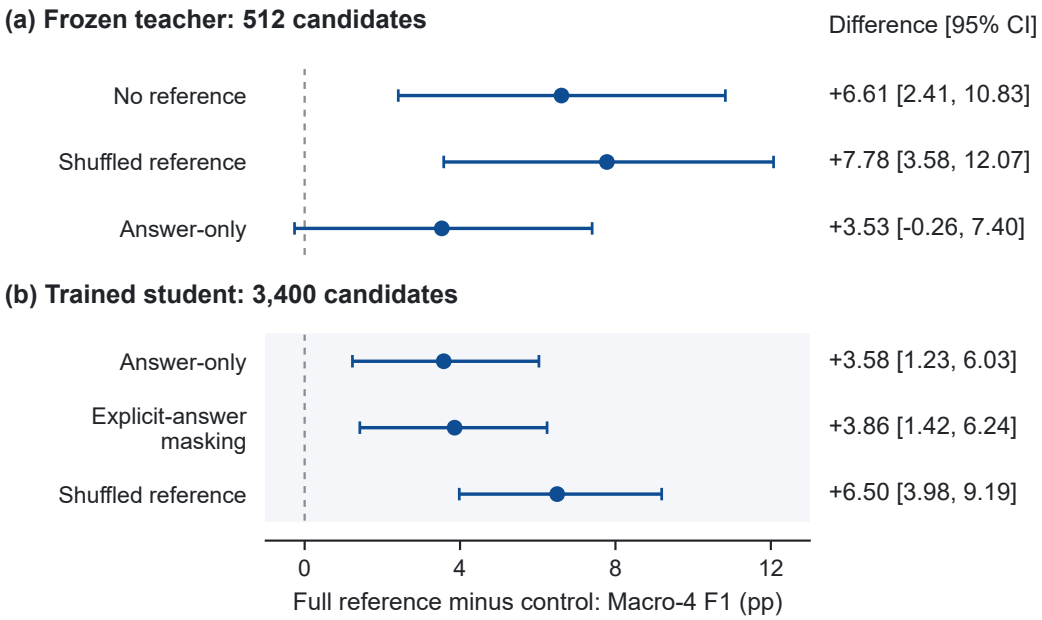


Figure 2: Reference effects in two complementary experiments. (a) Full reference minus each control for a frozen teacher on 512 candidates. (b) Full-reference training minus each control for the reference-free student on all 3,400 candidates. Points are Macro-4 F1 differences; bars are paired 95% problem-cluster intervals. The cohorts and interventions differ between panels.

Full references improve Macro-4 F1 over answer-only by 3.58 points $[1.23, 6.03]$ and over shuffled references by 6.50 points $[3.98, 9.19]$ (Figure 2b). Explicit-answer masking lowers F1 by 3.86 points $[1.42, 6.24]$. Together, the interventions identify value in both derivation content and problem correspondence. These effects are measured after training with references absent from the student's evaluation input.

5.4 Errors Concealed by Correct Final Answers

Table 2: Local judgments on correct-answer candidates with process errors. The full core group has 521 examples; its OlympiadBench and Omni-MATH portions contain 161 and 259. Reference ablations are OPSD-PRM variants. Exact denotes first-error localization; false acceptance denotes a valid -1. All rates are percentages; bold marks the best value per column.

	Core group ($n = 521$)		Core exact ↑	
Model / training reference	Exact ↑	False acceptance ↓	Olympiad	Omni
Qwen2.5-7B-Instruct	15.74	46.64	13.66	14.67
EurusPRM-Stage1 (local)	12.09	47.02	13.04	6.95
OPSD-PRM, full reference	**26.10**	**22.46**	**19.25**	**27.41**
OPSD-PRM: teacher-reference ablations				
Answer-only	20.54	36.66	14.91	22.39
Explicit-answer masking	22.07	33.78	14.91	21.24
Shuffled reference	18.23	45.68	11.80	19.69

The 521 correct-answer erroneous solutions directly test the example motivating this study. Full-reference training reaches 26.10% exact localization, compared with 20.54% for answer-only and 18.23% for shuffled references (Table 2). The paired gains are 5.57 points $[1.54, 9.68]$ and 7.87 points $[3.82, 11.81]$. The gain is present in exact error-index matches, as well as in rejection rates.

EurusPRM-Stage1 provides an external comparison: OPSD localizes 26.10% of errors versus its 12.09%, with false acceptance of 22.46% versus 47.02%. Full-reference training also lowers acceptance of fully correct solutions from answer-only's 72.77% to 59.03%. Benchmark F1 accounts for both localization and correct acceptance; Appendix C.3 gives the behavior counts.

The pattern extends to difficult subsets. Full rather than shuffled references improve F1 by 5.90 points $[1.42, 10.42]$ on OlympiadBench and 5.11 points $[1.14, 9.09]$ on Omni-MATH. Their core-group localization rates also improve (Table 2). Full–answer-only subset intervals are wider and cross zero; the complete contrasts are in Appendix C.4.

5.5 Candidate Composition and Supervision

We fix a separate pool of 1,024 problems, each with a successful and a failed candidate, and choose one candidate per problem at success proportions of 12.5%, 25%, and 50%. The successful-problem sets are nested. Both methods share the selected candidates, two visits per problem, 64 updates, and learning rate 5×10^{-6}. CE calibrates each threshold on GSM8K. The two endpoints are repeated with seed 20260915, keeping candidate identities and visit order fixed.

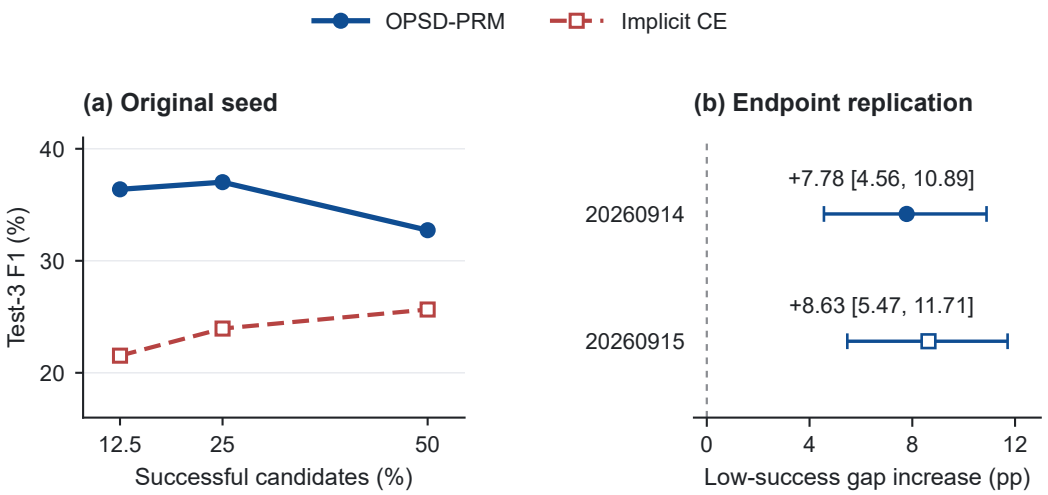


Figure 3: The low-success advantage is reproduced in a second training seed. (a) Original-seed F1 over three success proportions. (b) The increase in the OPSD–CE F1 gap from 50% to 12.5%, separately for two seeds, with paired evaluation-sample intervals. All configurations use 1,024 problems and 64 updates. At fixed total size, lower success proportion also means more failed candidates.

In the original seed, CE's F1 rises from 21.53 to 23.95 to 25.65 as the success proportion increases, while OPSD scores 36.38, 37.02, and 32.73. The low-success condition increases OPSD's relative advantage by 7.78 points $[4.56, 10.89]$. The second seed reproduces the direction with an 8.63-point increase $[5.47, 11.71]$ (Figure 3). On the 514 Test-3 correct-answer process errors, the corresponding localization interactions are 6.61 and 11.87 points, both with positive paired intervals (Appendix E).

Reference-guided training remains effective in these failure-heavy pools. The intervention establishes a candidate-composition effect: it jointly changes the numbers of successful and failed examples, while the questions and training budget remain fixed.

6 Training Budget and Distillation Choices

We examine candidate budgets and alternative training targets using shared candidate pools and reference-free evaluation.

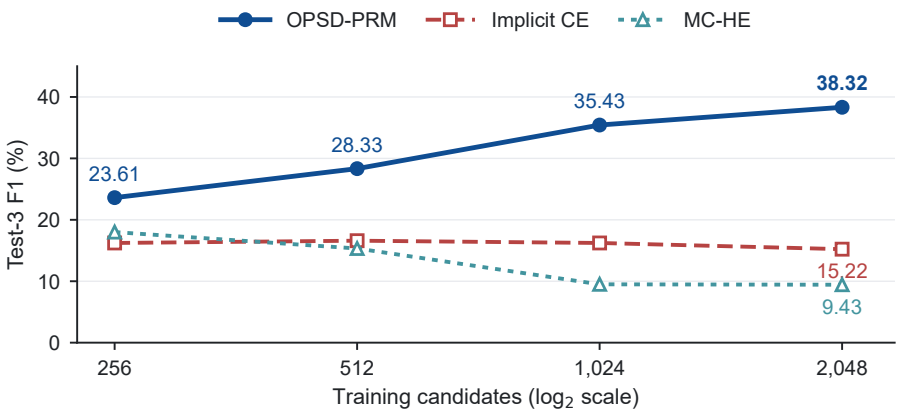


Figure 4: OPSD improves with candidate budget under a fixed two-epoch training rule. Candidates and visits are shared across methods at each scale, using one training seed and the original common learning rate. Test-3 excludes GSM8K calibration. Learning-rate-selected results at 1,024 candidates are reported separately in Table 3.

6.1 Candidate Budget and Learning-Rate Selection

Increasing the candidate budget raises OPSD from 23.61 to 38.32 Test-3 F1, above both scoring baselines at each evaluated scale (Figure 4). Each candidate is visited twice, so updates increase from 16 to 128 with dataset size. This is data utilization under a fixed access rule. The complete subset results appear in Appendix D.

At 1,024 candidates, each baseline selects its learning rate from $\{5 \times 10^{-6}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$ using GSM8K F1. Implicit CE improves from 16.23 to 18.92 Test-3 F1, while MC-HE retains 9.50. OPSD's fixed configuration scores 35.43. Appendix D.2 reports the full grid. The check addresses learning-rate sensitivity at this budget; longer-training optima are outside the fixed-access comparison.

6.2 Teacher-Evaluation SFT and the Pointwise Cap

Table 3: Test-3 F1 (%) with 1,024 candidates and 64 updates. CE/MC-HE use GSM8K-selected learning rates; reference-based controls use 5×10^{-6} and share the public prompt and greedy verdict interface.

Configuration	Test-3 F1
OPSD-PRM, pointwise cap	**35.43**
Implicit CE, selected learning rate	18.92
MC-HE, selected learning rate	9.50
Teacher-evaluation SFT	33.84
OPSD-PRM, ordinary forward KL	31.50

Teacher-SFT uses one format-valid reference-conditioned evaluation per candidate as a fixed hard-token target. Its F1 is 33.84, compared with OPSD's 35.43: a difference of 1.59 points $[-0.82, 3.97]$. Thus, reference-conditioned teacher text also trains an effective verifier; this comparison does not establish a reliable advantage for on-policy distribution targets. It varies both target form and prefix source.

Removing only the pointwise cap lowers F1 to 31.50. The capped objective gains 3.93 points $[1.49, 6.43]$ in this configuration. This supports its use for verification, alongside OPSD's original rationale for clipping (Zhao et al., 2026b). The capped loss remains a signed surrogate; Appendix A gives its gradient, and Appendix D reports the controls' full results and local judgments.

7 Related Work

Outcome supervision and implicit process rewards. Outcome verifiers learn from answer correctness (Cobbe et al., 2021; Hosseini et al., 2024); iterative reasoning training can also use successful generated solutions (Zelikman et al., 2022). Likelihood-ratio parameterizations connect preference objectives to implicit rewards (Rafailov et al., 2023; Ethayarajh et al., 2024; Chen et al., 2024). Token-level interpretations connect these objectives to local values (Rafailov et al., 2024); attention-based reward redistribution provides another route to dense feedback (Chan et al., 2024). Yuan et al. (2025) recover useful process rewards from outcome supervision, and PRIME uses implicit rewards for policy optimization (Cui et al., 2025). We study first-error judgments where final-answer correctness does not specify process validity, using reference interventions to examine the contribution of worked derivations.

Automatic and generative process supervision. Human annotations directly supervise intermediate judgments (Uesato et al., 2022; Lightman et al., 2024). Automatic alternatives obtain labels from continuations or search, as in Math-Shepherd and OmegaPRM (Wang et al., 2024; Luo et al., 2024), or estimate progress under a prover policy (Setlur et al., 2025). Step-DPO constructs stepwise preferences (Lai et al., 2024); SPARE produces discrete reference-guided step annotations (Rizvi et al., 2026). Generative verification expresses reward modeling through next-token prediction or evaluation rationales (Zhang et al., 2025a; Zhao et al., 2026a; Khalifa et al., 2026). Our experiments examine reference content and problem alignment within a fixed generative verifier; the same-interface SFT control compares two ways to transfer reference-informed evaluations.

Distillation with privileged information. Knowledge distillation transfers teacher predictions (Hinton et al., 2015), and privileged-information learning allows richer training inputs than deployment inputs (Vapnik & Izmailov, 2015). Learner-generated states support imitation learning (Ross et al., 2011); GKD studies student-generated language-model sequences, and MiniLLM uses reverse KL (Agarwal et al., 2024; Gu et al., 2024). OPSD combines reference conditioning and student sampling for reasoning generation (Zhao et al., 2026b). We use its released frozen-teacher option and pointwise cap to study fixed-candidate verification. The reference interventions examine what information this procedure transfers to local judgments.

8 Conclusion

Outcome fitting and first-error judgment impose different demands on supervision. The score-space analysis and checkpoint diagnostic make this distinction concrete. Reference interventions show that worked derivations improve independent verification, including localization of errors concealed by correct answers. The advantage over Implicit CE increases at low success proportions and repeats at the endpoints in two training seeds. OPSD implements the transfer; teacher SFT also trains an effective verifier. These findings concern mathematical first-error verification with one backbone and available references, without human candidate-step labels. Other domains and downstream reward uses remain natural extensions.

Reproducibility statement

Appendix A gives extended analysis and diagnostic details. Appendix B specifies data preparation, objectives, hyperparameters, prompts, and evaluation. Appendices C–E provide complete results and replication. The source package includes editable figures, plot data, and prompt/evaluation excerpts; model tensors and full server training assets are separate.

AI use statement

AI tools assisted with writing and polishing, literature retrieval, drafting the Introduction, Related Work, and appendix text, and mathematical analysis. The authors are responsible for the final text, references, and scientific claims.

REFERENCES

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

Alex James Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. Dense reward for free in reinforcement learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6136–6154. PMLR, 2024. URL https://proceedings.mlr.press/v235/chan24a.html.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456v1. Version 1.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12634–12651. PMLR, 2024. URL https://proceedings.mlr.press/v235/ethayarajh24a.html.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *International Conference on Learning Representations*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

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

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=FPVCb0WMuN.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs, 2024. URL https://arxiv.org/abs/2406.18629.

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

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision, 2024. URL https://arxiv.org/abs/2406.06592.

PRIME-RL. EurusPRM-Stage1, 2025a. URL https://huggingface.co/PRIME-RL/EurusPRM-Stage1. Model card.

PRIME-RL. EurusPRM-Stage2, 2025b. URL https://huggingface.co/PRIME-RL/EurusPRM-Stage2. Model card and ProcessBench evaluation protocol.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a Q-function. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kEVcNxtqXk.

Md Imbesat Hassan Rizvi, Xiaodan Zhu, and Iryna Gurevych. SPARE: Single-pass annotation with reference-guided evaluation for automatic process supervision and reward modelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32808–32816, 2026. doi: 10.1609/aaai.v40i39.40560.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015. URL https://jmlr.org/papers/v16/vapnik15b.html.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024. doi: 10.18653/v1/2024.acl-long.510.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73511–73525, 2025. URL https://proceedings.mlr.press/v267/yuan25c.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations*, 2025a.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-acl.547.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. GenPRM: Scaling test-time compute of process reward models via generative reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34932–34940, 2026a. doi: 10.1609/aaai.v40i41.40797. URL https://arxiv.org/abs/2504.00891v1. Extended version, Table 1.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2601.18734v3. Extended version, v3.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.50.

A Extended Analysis of Supervision

This appendix expands the response-level analysis in Section 2 and gives the gradient of the implemented distillation objective.

A.1 Shared Outer Weights for Response-Score Objectives

For fixed examples, $R_i = \beta \sum_t \log[B_\phi(y_{i,t} \mid h_{i,t})/C(y_{i,t} \mid h_{i,t})]$ with fixed C. If $\mathcal{L} = f(R_1, \ldots, R_m)$ contains no other dependence on ϕ, Equation 2 follows by the chain rule. Position-specific model gradients differ even when their outer coefficient is shared.

For binary CE,

$$\mathcal{L}_{\mathrm{CE}} = -l \log \sigma(R) - (1-l)\log(1-\sigma(R)), \qquad -\nabla_\phi \mathcal{L}_{\mathrm{CE}} = \beta[l - \sigma(R)] \sum_t \nabla_\phi \log B_\phi(y_t \mid h_t).$$

For a failed response, $l = 0$ and the coefficient is negative. Once its predicted success probability is small, the coefficient can also become small without an explicit requirement to resolve its internal first-error location.

For DPO, let $\Delta R = R^+ - R^-$ and $\mathcal{L}_{\mathrm{DPO}} = -\log \sigma(\Delta R)$ (Rafailov et al., 2023). Then

$$-\nabla_\phi \mathcal{L}_{\mathrm{DPO}} = \beta \sigma(-\Delta R) \left[\sum_t \nabla_\phi \log B_\phi(y_t^+ \mid h_t^+) - \sum_t \nabla_\phi \log B_\phi(y_t^- \mid h_t^-) \right].$$

Identical shared-prefix terms cancel exactly in DPO. Pair construction can therefore supply localization structure in addition to the sequence-level preference, while the parameter gradient at each remaining position depends on its context. KTO with a detached reference point and NCA also fit the response-score form, with different outer weights (Ethayarajh et al., 2024; Chen et al., 2024). A differentiable reference point or other terms outside the response scores require additional gradients.

The response-weighted form resembles a terminal-reward REINFORCE update (Williams, 1992). Here the coefficient comes from a supervised response objective; it need not have policy-gradient sampling or baseline properties.

A.2 Aggregate Constraints in Step-Score Coordinates

Treating step scores $\mathbf{s} = (s_1, \ldots, s_K)$ as free coordinates, write $R = \sum_k s_k$ and $\mathcal{L} = \ell(R, l)$. Any perturbation with $\sum_k \delta_k = 0$ leaves this single-candidate loss unchanged. For binary CE,

$$\nabla_s^2 \mathcal{L} = \sigma(R)(1 - \sigma(R))\mathbf{1}\mathbf{1}^\top.$$

The example in Section 2 illustrates these constant-total directions with different first-error predictions. This is a statement about the aggregate objective in free score coordinates. Language-model probabilities, shared parameters, and other training examples constrain realizable perturbations.

A.3 Prefix-Dependent Distribution Targets

Let a_{jv} be temperature-scaled logits, so $p_j = \mathrm{softmax}(a_j)$, and hold q_j fixed. For uncapped forward KL,

$$D_j = \sum_v q_j(v) \log \frac{q_j(v)}{p_j(v)}, \qquad \frac{\partial D_j}{\partial a_{ju}} = p_j(u) - q_j(u).$$

For the implemented cap, define $d_{jv} = q_j(v) \log[q_j(v)/p_j(v)]$ and $m_{jv} = \mathbf{1}[d_{jv} < \tau]$. Away from the cap boundary,

$$\frac{\partial \sum_v \min(d_{jv}, \tau)}{\partial a_{ju}} = p_j(u) \sum_v m_{jv} q_j(v) - m_{ju} q_j(u).$$

Gradients with respect to unscaled logits include $1/T$; the per-example average includes $1/|z|$. The cap changes the ordinary KL gradient and need not define a nonnegative divergence or guarantee exact teacher matching. We use the inherited OPSD surrogate with targets conditioned on the evaluation prefix and vocabulary item. The targets supervise evaluation tokens; the first-error experiments assess the resulting candidate-step judgments.

A.4 Low Outcome Residual and Unresolved Localization

We use the GSM8K-selected CE checkpoint from the original 1,024-candidate study (64 updates, learning rate 5×10^{-5}). For each of the 3,000 Test-3 candidates, cached full-precision terminal scores supply $p_{\text{out}} = \sigma(R)$ using $\beta = 0.05$ and the complete candidate-token likelihood-ratio sum. Prompt tokens and an additional EOS are excluded. The outcome label l is the benchmark's final-answer-correctness field. The residual is $|p_{\text{out}} - l|$; the primary condition is at most 0.10. All scores are available, so no new model forward pass is required. First-error predictions retain the frozen threshold 0.38369184592296146.

Table 4: Low-residual candidates with process errors. The condition is fixed before examining localization. Percentages use the reported group denominators; one independent example receives no interval.

Final answer	Total	Low residual	Share (%)	Location failures	Failure % [95% CI]
Correct	514	1	0.19	1	100.00 [NA]
Incorrect	1500	924	61.60	698	75.54 [72.65,78.30]

The 924 low-residual failed-answer candidates include 698 incorrect first-error predictions. Thus, a small outcome residual can coexist with unresolved localization. On this same CE-conditioned set, exact localization is 24.46% (226/924) for CE, 22.08% (204/924) for OPSD, and 19.37% (179/924) for Teacher-SFT. The latter two use the same 1,024-candidate budget. This diagnostic does not show that reference distillation repairs this particular subset. In the correct-answer error group, only one candidate meets the condition and none of the three models localizes it correctly.

This fixed-checkpoint diagnostic uses model sigmoid scores and makes no calibration assumption. It establishes coexistence of outcome fitting and localization errors, rather than a temporal account of how they were learned.

B Data, Training, and Evaluation Details

The experiments share a Qwen2.5-7B-Instruct backbone and use existing candidate solutions. This section specifies the data, supervision construction, training configuration, and evaluation rules.

B.1 Data Sources and Training Sets

The main model uses problems, pre-generated candidates, and native reference solutions from the OpenAI PRM800K phase-2 training release (Lightman et al., 2024). The source file is `prm800k/data/phase2_train.jsonl`, containing 97,782 records. This release uses an expanded MATH training split rather than only the original 7,500 MATH training problems.

The input fields are `question.problem`, `question.pre_generated_steps`, and `question.ground_truth_solution`. Candidates were generated upstream by the PRM800K release's GPT-4-family generator; references are the dataset's native MATH solutions (Hendrycks et al., 2021). Distillation uses the frozen Qwen2.5-7B-Instruct backbone as teacher. Human step ratings and verifier scores are excluded from sample selection and training-target construction; references enter the teacher context only.

Preparation excludes missing or empty candidates/references, deduplicates complete problem–candidate pairs, and removes normalized exact problem matches with ProcessBench, MATH500, and the PRM800K test splits. Normalization removes whitespace and ignores case. The retained pool contains 69,072 distinct candidates. A fixed problem-level split separates internal training and development examples; a length filter requires both public and privileged prompts to fit 3,072 tokens. The prepared training pool contains 9,999 candidates from 5,560 problems. A fixed permutation supplies 6,400 distinct candidates from 4,247 problems for the 200-update main model. The normalized exact intersection with each ProcessBench subset is zero. Internal development examples are separate from the ProcessBench evaluation data.

The four data-efficiency subsets select distinct problems in a deterministic order, one candidate per problem, with nested sizes. All methods use the same candidates and visit order at a given size. Selection does not balance candidate outcomes; unparseable final answers remain failed outcomes. The additional length filter requires the formatted candidate solution sequence to fit 3,072 tokens.

Table 5: Nested data-efficiency training sets. MC continuations are eight per nonterminal prefix. Each candidate is visited twice.

Problems	Updates	Successful	Failed	Steps	MC continuations
256	16	34	222	3393	25096
512	32	62	450	6722	49680
1024	64	133	891	13613	100712
2048	128	248	1800	27085	200296

The candidate-composition study uses a separate set of 1,024 problems for which both a successful and a failed candidate are available. Its selection rule differs from the data-efficiency study. The within-pool intervention, detailed in Appendix E, changes which candidate is used for some problems while retaining the same problem set.

B.2 Training Configuration

Table 6 gives the training configuration. Every arm initializes a fresh LoRA adapter on the original backbone; the initial LoRA B matrices are zero and only adapter parameters are trainable. Each experiment uses its own complete linear-decay schedule, including the 64-update learning-rate comparison.

Table 6: Common training configuration. Main/reference-control runs use 200 updates; scaling and composition runs decay the learning rate over their own full update budgets.

Setting	Value
Backbone / frozen teacher	Qwen2.5-7B-Instruct
Adapter	LoRA rank 64, alpha 128, dropout 0, no bias
Target projections	`q,k,v,o,gate,up,down` projections
Global batch	4 devices $\times$ microbatch 1 $\times$ accumulation 8
Optimizer	AdamW; betas $(0.9, 0.999)$; epsilon 10^{-8}
Weight decay / warmup	0 / 0
Initial learning rate	5×10^{-6}, linear decay
Maximum gradient norm	0.1
Precision	BF16 backbone; FP32 log-probability/loss operations
Hardware	4 $\times$ NVIDIA A800 80GB
OPSD evaluation sampling	$T = 1.1$, top-$p = 0.95$, top-$k = 20$
Prompt / completion / context	3,072 / 1,024 / 4,096 tokens
Distribution temperature / cap	1.1 / 0.05
Additional anchor loss	None
Scaling seed / composition endpoint seeds	20260914 / 20260914, 20260915

B.3 OPSD Implementation

The teacher forward disables the student adapter and uses the initial backbone. The sampled evaluation token IDs are identical between teacher and student, with next-token positions aligned after their different prompt lengths. Both log-softmax operations use temperature 1.1; no additional T^2 factor is applied. Vocabulary contributions are capped, summed, averaged over valid evaluation tokens within each example, and then averaged across examples. Prompt and padding tokens are masked.

The pointwise cap follows the official OPSD implementation. In its forward-KL configuration, the code applies `clamp(max=token_clip)` to signed vocabulary contributions before reduction; our configuration uses $\tau = 0.05$. Version 3 of OPSD motivates this rule by large stylistic-token

contributions and reports a clipping ablation for reasoning generation (Zhao et al., 2026b). The capped surrogate can be negative; Appendix A distinguishes its gradient from ordinary KL.

The student-prefix design follows the learner-distribution principle in imitation learning (Ross et al., 2011) and on-policy language-model distillation (Agarwal et al., 2024): teacher supervision is evaluated on prefixes generated by the learner.

The main 200-update model generates online evaluations with colocated vLLM (tensor parallelism one, GPU-memory fraction 0.3) and refreshes student adapters at every update. Scaling and composition experiments use Transformers generation with a per-request KV cache, BF16/SDPA training, and no gradient checkpointing. Generation uses EOS token IDs 151645 and 151643. The matched experiments use fused AdamW and DDP with four devices. The mathematical objective is shared across these implementations.

B.4 Implicit CE

The public format is a Qwen chat problem followed by an assistant continuation with numbered `Step K` segments. A frozen original backbone supplies reference log-probabilities. The response score is the sum of candidate-token log-probability differences scaled by $\beta = 0.05$; prompt tokens are excluded. Binary cross-entropy with logits trains this score against the automatically checked final-answer outcome.

At evaluation, the score at each step boundary is the sigmoid of the cumulative likelihood-ratio sum through that boundary. The first score below the checkpoint's calibrated threshold gives the first-error prediction; otherwise the candidate is accepted. Candidate token fragments are appended as strict prefixes.

B.5 Math-Shepherd-Style MC-HE

For each nonterminal candidate prefix, the frozen backbone samples eight continuations with temperature 0.7, top-$p = 0.95$, and a 2,048-token limit. The prefix receives a positive hard-estimation label if at least one continuation has a correct final answer, and a negative label otherwise (Wang et al., 2024). The terminal step uses the original candidate outcome. Failed and truncated continuations stay in the sample set. Generation uses vLLM with shared-prefix caching.

The final training/readout interface follows the Math-Shepherd step-marker convention. A Cyrillic step marker with token ID 16748 is appended at each step boundary; its two code points are U+043A and U+0438. At the marker position, full-vocabulary cross-entropy targets the existing positive or negative token, with IDs 10 and 12. Previous gold labels are not inserted into the input. Step losses are averaged within each candidate, followed by candidate averaging. At evaluation the marker-position score is $P(+)/(P(+) + P(-))$. The first score below the calibrated threshold is the predicted error. This implementation uses continuation sampling and step classification, without tree search or a new classification head.

B.6 Generated Verdicts and Score Calibration

ProcessBench provides first-error annotations for candidate solutions associated with GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), and Omni-MATH (Gao et al., 2025). We use the ProcessBench candidate solutions and annotations, rather than treating the original benchmark answer labels as process labels.

The main student and its reference ablations use the same ProcessBench inputs, chat instruction, greedy decoding, and parser. Candidate paragraphs are numbered from zero. The public instruction asks for ordered verification followed by a boxed first-error index or -1; its additional verification sentence is *Before giving the boxed index, write a concise verification of the solution in paragraph order.* The teacher inserts the original reference before the candidate solution in its privileged context. The student and all ordinary benchmark evaluation inputs omit references.

Greedy evaluation uses one continuation, temperature zero, top-$p = 1$, seed 42, and at most 1,024 new tokens. The parser reads the content of the final matched boxed expression and converts it to an integer. If that final content is not an integer, it does not fall back to an earlier verdict. A missing,

malformed, or out-of-range verdict is invalid; it remains in its annotated group's denominator and counts as unsuccessful.

Main-model evaluations use vLLM 0.10.2, V1 disabled, BF16, tensor parallelism one, eager execution, context length 4,096, and no prefix caching. New OPSD evaluations likewise disable prefix caching and use four independent single-GPU workers. Score models use Transformers BF16/SDPA forward passes. Every new score-model checkpoint chooses its own threshold from $i/1999$, $i = 0, \ldots, 1999$, by GSM8K F1, with smallest-threshold tie breaking. The selected threshold is held fixed on Test-3.

The public EurusPRM-Stage1 comparison uses its recorded Qwen2.5-Math-7B-Instruct likelihood-ratio reference (PRIME-RL, 2025a). The readout follows the released ProcessBench protocol: one-based `Step K` formatting, cumulative likelihood-ratio scale 0.001, sigmoid, and threshold 0.5015 (PRIME-RL, 2025b). Its step index is mapped to the zero-based benchmark index. This local Stage1 run supplies the paired diagnostic predictions; Table 1 uses the separately reported GenPRM scores.

B.7 Statistical Procedures and Available Materials

Point estimates use each experiment's full specified evaluation set. Macro-4 and Test-3 average subset F1, while pooled localization and acceptance use their annotated-group denominators. Contrasts use unrounded statistics, so subtracting displayed rounded values can give a slightly different difference.

Paired bootstrap sampling clusters by normalized problem within each subset, retaining associated candidates together. The same draw is used for every compared model; subset F1 is recomputed before macro averaging. Composition interactions share draws across both methods and endpoint proportions. We use 5,000 draws, seed 42, percentile 95% intervals, and fixed thresholds. Subgroup contrasts are unadjusted for multiple comparisons. These intervals describe evaluation-sample uncertainty for the trained checkpoints and selected candidates. Main-model training variation is summarized by the three-seed sample standard deviation; composition endpoint results are listed separately for each training seed.

Appendix B.8 gives the complete prompt templates and core evaluation code. The accompanying manuscript source includes these excerpts, numerical figure data, and plotting code. The experimental records contain saved token inputs, visit orders, reference mappings, and per-example predictions. Model tensors and the full server execution assets are separate from this manuscript source package.

B.8 Prompt Templates and Reproducibility Excerpts

This section gives the complete public prompt, the teacher's reference insertion, and the core parsing and metric procedures. The source package includes the corresponding plain-text templates and code excerpts. Line wrapping in printed templates is for display; the plain-text files preserve their message content.

B.8.1 Public Student and Reference-Conditioned Teacher

The system message for both roles is:

```
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.
```

The student receives the following user message. The field `problem` is the original question; `tagged_response` contains the complete candidate, with each original step enclosed in zero-based paragraph tags. Qwen's native chat template supplies the assistant generation prefix.

```
The following is a math problem and a solution (split into paragraphs, enclosed with
tags and indexed from 0):

[Math Problem]

{problem}
```

```
[Solution]

{tagged_response}

Your task is to review and critique the solution paragraph by paragraph. Once you
identify an error in a paragraph, return the index of the paragraph where the earliest
error occurs. Otherwise, return the index of -1 (which typically denotes "not found").

Please put your final answer (i.e., the index) in \boxed{}.

Before giving the boxed index, write a concise verification of the solution in paragraph
order.
```

For example, the candidate is formatted as:

```
<paragraph_0>
{first original candidate step}
</paragraph_0>
<paragraph_1>
{second original candidate step}
</paragraph_1>
...
```

The teacher uses the same public template, inserting the following block immediately before `[Solution]`. Its `ground_truth_solution` is the original worked reference, including any final answer in the derivation.

```
**A correct solution to this problem is given below for your reference:**
<solution>
{question.ground_truth_solution}
</solution>
```

Both distributions are computed along the same student-generated evaluation token IDs. The teacher supplies next-token probabilities by forward evaluation; it does not generate a separate target evaluation in OPSD training.

B.8.2 Reference Interventions

Full-reference training preserves the original reference. Answer-only replaces the content inside the solution tags by a boxed native final answer, keeping the surrounding instructions unchanged. Shuffling uses a fixed donor mapping over the same 6,400 visits, with a different problem as donor and the reference multiset preserved. Masking replaces balanced boxed or fboxed blocks by a literal ellipsis command while retaining the rest of the reference. These operations change the teacher context; the student's input and evaluation prompt remain public and reference-free. Appendix C gives the intervention results and the remaining answer occurrences after explicit-answer masking.

B.8.3 Answer Checking and Verdict Parsing

Automatic outcome labels use the native final-answer fields and `math-verify` 0.8.0. Missing, unparseable, and timed-out candidate answers count as failures. MC labels apply the same checker to the full candidate prefix plus continuation. A standard answer that cannot be parsed stops preparation. The implementation excerpt is:

```
from math_verify import parse, verify, LatexExtractionConfig

def math_answer(s):
    return parse(
        '$' + s + '$', extraction_config=[LatexExtractionConfig()],
        parsing_timeout=2,
    )

def compare_answer(gold, prediction):
    g = math_answer(gold); p = math_answer(prediction)
    return dict(
        correct=bool(g and p and verify(g, p, timeout_seconds=2)),
        gold_parsed=bool(g), prediction_parsed=bool(p),
    )

def grade_continuation(gold, text):
    g = math_answer(gold)
    # The parser prioritizes the final boxed answer.
```

```
    p = parse(text, parsing_timeout=2)
    return dict(
        correct=bool(g and p and verify(g, p, timeout_seconds=2)),
        gold_parsed=bool(g), prediction_parsed=bool(p),
    )
```

Generated evaluation outputs are parsed with the following rule. Paragraph indices are zero-based; -1 accepts the entire candidate. The final matched boxed content determines the verdict, including when an earlier boxed expression was valid.

```
matches = re.findall(r'\\boxed\{([^}]*)\}', text)
try:
    pred = int(matches[-1].strip())
except (ValueError, IndexError):
    pred = None
if pred is not None and not -1 <= pred < len(steps):
    pred = None
```

The score-model readout returns the first step whose score is strictly below the calibrated threshold, or -1 if all steps pass. Calibration uses the 2,000-point grid and tie-breaking rule in Appendix B.

B.8.4 Metric Computation and Prediction Records

Let $\mathcal{E} = \{i : g_i \geq 0\}$ and $\mathcal{C} = \{i : g_i = -1\}$. On each subset,

$$E = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} \mathbf{1}[\hat{g}_i = g_i], \qquad C = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \mathbf{1}[\hat{g}_i = -1], \qquad F1 = \frac{2EC}{E + C}.$$

The score is zero when $E + C = 0$. An invalid prediction contributes zero to its group's numerator and remains in the denominator. Macro-4 averages the four subset F1 values; Test-3 averages MATH, OlympiadBench, and Omni-MATH. The correct-answer erroneous group is defined by ground-truth labels before evaluating predictions.

Prediction records contain subset, candidate ID, problem identity, gold first-error index, final-answer correctness, predicted index, and parsing status. Scoring methods also retain their step scores and threshold. The pair of subset and candidate ID identifies a record, while problem identity defines bootstrap clusters. An absent verdict is stored as null, distinct from the valid -1 verdict. The source package provides the metric function used to aggregate these fields; Appendix B.7 specifies the paired resampling procedure.

B.8.5 Run Environment and Material Organization

The reported runs use Linux and Python 3.10, with PyTorch 2.8.0, Transformers 4.57.1, PEFT 0.17.1, Accelerate 1.11.0, TRL 0.26.0, and vLLM 0.10.2. Training uses BF16 with FP32 log-softmax operations. The source package records the remaining dependency versions, complete prompts, core objective and evaluation excerpts, and aggregate result tables. Model tensors and the full server execution assets are separate from these manuscript materials.

C Reference Information and Local Judgments

We report the frozen-teacher reference contrast, the trained-student interventions, and their outcome-stratified and difficult-subset results.

C.1 Frozen-Teacher Diagnostic

The diagnostic uses 128 distinct problems per subset, each with one candidate, an available native reference, and original reference conditions fitting 2,560 prompt tokens. A deterministic ordering selects candidates without process-error labels or model outputs. References come from the original GSM8K answer derivation, MATH solution, OlympiadBench English mathematics solution, and Omni-MATH solution. Shuffling permutes references without fixed points within each subset, approximately matching lengths.

The untrained reference-free base samples one evaluation per candidate with temperature 1.1, top-$p = 0.95$, top-$k = 20$, and a 1,024-token limit, using seed $20260913 + i$. The shared prefix is

the first half of the original tokens before the first literal boxed marker (or EOS), rounded down and limited to 512 tokens. The exact IDs precede each reference-conditioned continuation. Greedy continuation adds at most 1,024 tokens, seed 42, with the ordinary boxed parser. The answer-only extension reuses all 512 candidates and prefixes, replacing the solution block by the boxed native final answer; it adds 512 continuations without retries. It uses GSM8K's last answer field after the four-hash delimiter, MATH's answer, OlympiadBench's first final-answer entry, and Omni-MATH's answer.

Table 7: Frozen-teacher F1 (%) under four reference conditions. Each subset has 128 examples. Erroneous/fully correct counts are 70/58, 70/58, 80/48, and 99/29 respectively.

Reference	GSM8K	MATH	Olympiad	Omni	Macro-4
None	48.80	30.56	27.51	**21.54**	32.10
Full matched	**64.63**	**37.90**	**30.88**	21.45	**38.71**
Shuffled	44.09	32.34	27.51	19.77	30.93
Answer-only	62.05	36.28	27.51	14.91	35.19

Table 8: Frozen-teacher judgments on the 63 correct-answer candidates with process errors, and paired full-reference contrasts. Rates are percentages; differences are pp.

Reference	Exact first error	False acceptance
None	17.46	52.38
Full matched	15.87	53.97
Shuffled	19.05	53.97
Answer-only	14.29	58.73
Full − answer-only [95% CI]	$+1.59\ [0.00, 5.17]$	$-4.76\ [-13.11, 3.12]$

Full-reference Macro-4 gains are 6.61 $[2.41, 10.83]$ over no reference, 7.78 $[3.58, 12.07]$ over shuffled references, and 3.53 $[-0.26, 7.40]$ over answer-only. The core group is small: full references yield 10 exact judgments versus 9 for answer-only and 11 for no reference. The teacher's aggregate benefit therefore does not establish an improvement in this particular subgroup. The teacher and student interventions use different cohorts and measure immediate conditioning and training effects, respectively.

The original diagnostic used 512 prefix samples and 1,536 condition continuations (732,843 generated tokens, 0.4107 allocated GPU-hours inside the workers). The answer-only extension adds 512 continuations; its additional cost is not included in that original total. All four conditions use the frozen base without LoRA.

C.2 Student Reference Interventions

All reference-control students use the 200-update configuration. Full references preserve the original worked solution. Answer-only provides the final answer without the derivation. Explicit-answer masking removes boxed/fboxed spans while retaining the surrounding reference. The recorded masking operation removes 6,505 such blocks; the answer literal still appears elsewhere in 2,021 of the 6,400 visits. Some removed blocks also contain calculations. The intervention therefore masks explicit answer spans while retaining any answer information elsewhere in the derivation.

The shuffled condition applies a fixed one-to-one permutation across training visits, always assigning a reference from a different problem. It preserves the complete reference-text multiset. All trained students use the same public evaluation prompt without a reference.

Table 9: OPSD-PRM teacher-reference ablations: student F1 (%). All variants are evaluated without references. Bold marks the best score per column.

OPSD teacher reference	GSM8K	MATH	Olympiad	Omni	Macro-4
Full reference	**52.94**	**43.28**	**30.93**	**34.39**	**40.38**
Answer-only	47.95	38.32	28.27	32.68	36.81
Explicit-answer masking	48.05	39.96	29.52	28.59	36.53
Shuffled reference	44.57	36.64	25.03	29.28	33.88

C.3 Outcome-Stratified Judgments

An erroneous solution with first-error index $g \geq 0$ has five mutually exclusive prediction categories: an earlier valid index ($0 \leq \hat{g} < g$), the exact index, a later valid index, explicit acceptance ($\hat{g} = -1$), or an invalid output. False acceptance counts only the fourth category. Invalid outputs remain a separate unsuccessful category. Step-level annotations are used through the first error only.

Table 10: Gold-label strata used by the local analyses.

Subset	Total	Correct answer, process error	Wrong answer, process error	Fully correct
GSM8K	400	7	200	193
MATH	1000	94	500	406
OlympiadBench	1000	161	500	339
Omni-MATH	1000	259	500	241
Total	3400	521	1700	1179

Table 11: Exhaustive prediction counts on the 521 correct-answer solutions with process errors. Reference interventions are OPSD-PRM variants.

Model	Earlier	Exact	Later	Accepted	Invalid
OPSD-PRM	84	136	136	117	48
Answer-only	52	107	144	191	27
Shuffled reference	47	95	102	238	39
Explicit-answer masking	71	115	112	176	47
EurusPRM-Stage1	55	63	158	245	0

Table 12: Pooled diagnostic rates (%). Pooled groups differ from the macro-averaged benchmark score. Reference interventions are OPSD-PRM variants. Bold marks higher exact localization or correct acceptance, and fewer invalid outputs.

Model	All errors: exact	Fully correct: accepted	All solutions: invalid
OPSD-PRM	**28.01**	59.03	7.74
Answer-only	22.78	72.77	4.82
Shuffled reference	19.68	**80.83**	6.03
Explicit-answer masking	22.29	72.35	8.82
EurusPRM-Stage1	19.95	65.73	**0.00**

Table 13: Full-reference minus each control: paired differences (pp) and 95% intervals. Core denotes correct-answer solutions with process errors. Reference interventions are OPSD-PRM variants.

Control	Macro-4 F1	Core exact	Core false acceptance
Answer-only	$+3.58$ $[+1.23, +6.03]$	$+5.57$ $[+1.54, +9.68]$	-14.20 $[-19.08, -9.52]$
Shuffled reference	$+6.50$ $[+3.98, +9.19]$	$+7.87$ $[+3.82, +11.81]$	-23.22 $[-28.18, -18.29]$
Explicit-answer masking	$+3.86$ $[+1.42, +6.24]$	$+4.03$ $[+0.18, +8.05]$	-11.32 $[-15.80, -6.64]$
EurusPRM-Stage1	$+5.86$ $[+2.92, +8.65]$	$+14.01$ $[+9.36, +18.69]$	-24.57 $[-30.29, -18.99]$

C.4 Difficult-Subset Results

The following tables extend the difficult-subset columns of Table 2 with F1 and reference-intervention intervals.

Table 14: Difficult-subset results. F1 and exact localization of correct-answer process errors (%). Reference interventions are OPSD-PRM variants.

Model	Olympiad F1	Olympiad exact	Omni F1	Omni exact
OPSD-PRM	**30.93**	**19.25**	**34.39**	**27.41**
Answer-only	28.27	14.91	32.68	22.39
Shuffled reference	25.03	11.80	29.28	19.69
Explicit-answer masking	29.52	14.91	28.59	21.24
EurusPRM-Stage1	24.70	13.04	17.53	6.95

Table 15: Difficult-subset reference effects (pp), with paired 95% intervals. Reference interventions are OPSD-PRM variants.

Control	Subset	F1 difference	Core exact difference
Answer-only	Olympiad	$+2.66$ $[-1.40, +6.94]$	$+4.35$ $[-2.24, +11.04]$
Answer-only	Omni	$+1.71$ $[-2.05, +5.66]$	$+5.02$ $[-1.25, +11.25]$
Shuffled	Olympiad	$+5.90$ $[+1.42, +10.42]$	$+7.45$ $[+0.62, +14.71]$
Shuffled	Omni	$+5.11$ $[+1.14, +9.09]$	$+7.72$ $[+1.62, +13.67]$
Explicit-answer masking	Olympiad	$+1.41$ $[-2.50, +5.55]$	$+4.35$ $[-2.94, +11.84]$
Explicit-answer masking	Omni	$+5.80$ $[+1.98, +9.63]$	$+6.18$ $[+0.41, +11.72]$

The pooled results distinguish exact localization, acceptance, and invalid outputs. The full-reference model has 263 invalid verdicts: 48 in the correct-answer erroneous group, 152 in the wrong-answer erroneous group, and 63 among fully correct solutions. All contribute to the reported denominators.

C.5 Examples of Generated Process Judgments

These examples come from saved main-model evaluations. For each behavior, we take the first matching case in the fixed ProcessBench order. The passages below quote the relevant candidate and evaluation excerpts; gold indices identify the benchmark's first error.

Accurate localization: GSM8K-0. The question describes 18 pink flamingos, six of which are painted white and returned, followed by 18 additional pink flamingos. Candidate paragraph 1 incorrectly concludes: "by the end of Saturday, Sue has $12 + 6 = 18$ pink flamingos and 6 white flamingos." The model identifies the color-counting error: "Sue should have 12 pink flamingos and 6 white ones after painting and replacing them." Its verdict is $\boxed{1}$, matching the gold first-error index. The evaluation distinguishes a correct subtraction from the subsequent invalid interpretation.

Premature error: GSM8K-6. Doing one's own taxes costs three hours of freelance work at \$35 per hour; an accountant charges \$90. Candidate paragraph 1 correctly computes \$105−\$90=\$15.

The first error is in paragraph 2, which reverses who benefits before returning a negative answer. The model instead disputes the interpretation of the correct calculation in paragraph 1 and returns $\boxed{1}$ rather than the gold $\boxed{2}$. This case illustrates a valid prefix rejected because of an unnecessary distinction in wording.

Missed error: GSM8K-1. The question states that Cindy's math and science books weigh two pounds each. Candidate paragraph 1 invents "2 math books and 2 science books", then calculates eight pounds for these books. The model repeats the invented multiplicity in its evaluation and states "There are no issues here." It accepts the solution with $\boxed{-1}$, while the gold first-error index is 1. The arithmetic is internally consistent after the unsupported assumption, but the assumption itself is missed.

D Training Routes and Candidate Budgets

This section reports fixed-budget comparisons, baseline learning-rate selection, and the reference-conditioned training controls.

D.1 Candidate-Budget Results

Table 16 reports all four subsets for Figure 4. Every scale uses the shared 5×10^{-6} learning rate and two visits per candidate. Learning-rate selection below concerns the 1,024-candidate budget.

Table 16: Complete data-efficiency results. F1 (%); GSM8K is the calibration set for Implicit CE and MC-HE. All checkpoints are the final two-epoch checkpoints. Bold marks the best score at each data scale.

N	Method	GSM8K	MATH	Olympiad	Omni	Test-3
256	OPSD-PRM	**36.56**	**30.37**	**18.52**	**21.93**	**23.61**
256	Implicit CE	20.17	18.05	15.14	15.51	16.23
256	MC-HE	21.45	20.20	16.64	17.16	18.00
512	OPSD-PRM	**47.66**	**36.59**	**22.37**	**26.04**	**28.33**
512	Implicit CE	18.67	17.93	15.06	16.79	16.60
512	MC-HE	20.58	14.53	14.44	17.01	15.33
1024	OPSD-PRM	**44.82**	**40.15**	**31.94**	**34.20**	**35.43**
1024	Implicit CE	21.29	17.74	14.17	16.79	16.23
1024	MC-HE	19.93	11.61	8.40	8.50	9.50
2048	OPSD-PRM	**50.35**	**44.59**	**35.04**	**35.32**	**38.32**
2048	Implicit CE	28.79	19.00	12.44	14.21	15.22
2048	MC-HE	16.95	14.32	2.65	11.34	9.43

D.2 Learning-Rate Selection at 1,024 Candidates

We retain the 1,024-candidate pool, 2,048 accesses, 64 updates, global batch 32, and seed 20260914. Only the baseline learning rate varies over $\{5 \times 10^{-6}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$. Candidate order, MC labels, fresh-adapter initialization, LoRA, loss, optimizer, and the 64-step linear decay are fixed. OPSD uses its original 5×10^{-6} checkpoint.

GSM8K F1 selects the learning rate and the threshold for each baseline, with ties resolved toward smaller values. The selected new checkpoint is then evaluated on Test-3; the original checkpoint's recorded results are also reported. Table 17 gives the complete development grid, and Table 18 gives the test results.

Table 17: Complete development grid. GSM8K is used for learning-rate selection and threshold calibration. Bold marks the selected row's F1 within each method.

Method	Learning rate	GSM8K F1 (%)	Threshold
Implicit CE	5×10^{-6}	21.29	0.4057028514
Implicit CE	2×10^{-5}	35.08	0.4402201101
Implicit CE	5×10^{-5}	**45.55**	0.3836918459
MC-HE	5×10^{-6}	**19.93**	0.1195597799
MC-HE	2×10^{-5}	11.04	0.2451225613
MC-HE	5×10^{-5}	14.66	0.2231115558

Table 18: Learning-rate check: per-subset F1 (%). Test-3 averages MATH, OlympiadBench, and Omni-MATH F1. Bold marks the highest F1 per column. MC-HE selects its original configuration.

Method	Config.	GSM8K	MATH	Olympiad	Omni	Test-3
OPSD-PRM	Fixed	44.82	**40.15**	**31.94**	**34.20**	**35.43**
Implicit CE	Original	21.29	17.74	14.17	16.79	16.23
Implicit CE	Selected	**45.55**	24.27	13.01	19.47	18.92
MC-HE	Original / selected	19.93	11.61	8.40	8.50	9.50

D.3 Teacher-Evaluation SFT and Uncapped Distillation

Both controls use the original 1,024-problem pool (133 successful and 891 failed candidates), seed 20260914, the same two-visit order, and 64-update linear decay from 5×10^{-6}. Teacher-SFT uses the frozen original backbone with complete references to generate one fixed evaluation target per candidate. Sampling uses temperature 1.1, top-$p = 0.95$, top-$k = 20$, and at most 1,024 tokens. Candidates are retained while sampling continues until a nonempty, naturally terminated, budget-valid completion has a legal boxed verdict. This checks format, not agreement with human first-error labels.

Target preparation takes 1,314 attempts for 1,024 accepted targets; 146 candidates fail on the first attempt and the maximum is 18 attempts. Each attempt's seed combines 20260914 with the recorded deterministic candidate/attempt identifier. The student receives only public inputs. Completion-only full-vocabulary CE is averaged within each example, including its generated EOS, then across examples. Targets are visited twice, totaling 1,259,710 supervised-token visits. Target generation costs 0.5078 GPU-hours, formal training 0.2951, and full evaluation 1.6475; a separate two-step smoke check uses 0.0176.

The uncapped control shares the initial adapter, candidate order, complete references, completion masks, distribution temperature, and normalization with capped OPSD. It removes only the vocabulary-term clamp and uses ordinary forward KL. Student evaluation samples can diverge as the trained models diverge. Both controls retain the ordinary public prompt, greedy generation budget, and verdict parser.

Table 19: Complete reference-conditioned route results at 1,024 candidates and 64 updates. F1 (%); core denotes the 514 Test-3 correct-answer process errors.

Configuration	GSM8K	MATH	Olympiad	Omni	Test-3	Core exact	Core accept
OPSD, capped	44.82	40.15	**31.94**	**34.20**	**35.43**	**24.32**	34.24
Teacher-SFT	**48.06**	**43.76**	29.26	28.50	33.84	23.74	**20.82**
OPSD, uncapped	47.59	41.36	26.87	26.26	31.50	21.98	22.57

Table 20: Paired route contrasts (pp), with 95% evaluation-sample intervals. Positive F1 or localization favors capped OPSD; positive false acceptance favors the control.

Capped OPSD minus	Test-3 F1	Core exact	Core false acceptance
Teacher-SFT	$+1.59\ [-0.82, 3.97]$	$+0.58\ [-3.75, 4.89]$	$+13.42\ [9.18, 17.83]$
Uncapped OPSD	$+3.93\ [1.49, 6.43]$	$+2.33\ [-1.63, 6.32]$	$+11.67\ [7.16, 16.10]$

Teacher-SFT and uncapped OPSD accept fewer core-group errors, while capped OPSD has the highest aggregate F1. Its localization differences in this subgroup have intervals containing zero. The SFT contrast changes hard/soft targets and teacher/student evaluation prefixes together. The cap contrast identifies an aggregate benefit at this budget without requiring the capped surrogate to share ordinary KL's mathematical properties.

D.4 Benchmark Context and Supervision Resources

Table 22 separates supervision information from the resources used to obtain it. Our experiments share the original backbone and candidate pool, then construct supervision through each route. Released checkpoints retain their source-specific data and evaluation protocols.

Table 21: Complete benchmark context accompanying Table 1. ProcessBench F1 (%). Bold marks the best score within each block; published rows retain their source protocols. †: Zheng et al. (2025, Table 3); ‡: Zhao et al. (2026a), arXiv v1, Table 1. Published critics use eight-sample voting; OPSD uses greedy decoding. Macro-4 averages the four subsets.

Model	GSM8K	MATH	Olympiad Bench	Omni MATH	Macro-4
Prompted language-model critics (7–8B)					
Meta-Llama-3-8B-Instruct†	13.1	13.8	4.8	12.6	11.1
Llama-3.1-8B-Instruct†	10.9	5.1	2.8	1.6	5.1
Qwen2.5-Math-7B-Instruct†	26.8	25.7	14.2	12.7	19.9
Qwen2.5-Coder-7B-Instruct†	14.3	6.5	4.1	1.8	6.7
Qwen2-7B-Instruct†	8.4	19.0	14.7	12.1	13.6
Qwen2.5-7B-Instruct†	**36.5**	**36.6**	**29.7**	**27.4**	**32.6**
Other process reward models					
Qwen2.5-Math-7B-PRM800K†	68.2	**62.6**	**50.7**	**44.3**	**56.5**
Skywork-1.5B†	59.0	48.0	19.3	19.2	36.4
Skywork-7B†	**70.8**	53.6	22.9	21.0	42.1
EurusPRM-Stage2‡	47.3	35.7	21.2	20.9	31.3
Qwen2.5-Math-7B-Math-Shepherd‡	62.5	31.6	13.7	7.7	28.9
Math-Shepherd-7B†	47.9	29.5	24.8	23.8	31.5
RLHFlow-Mistral-8B†	50.4	33.4	13.8	15.8	28.4
RLHFlow-Deepseek-8B†	38.8	33.8	16.9	16.9	26.6
Implicit baseline and reference-distilled verifier					
EurusPRM-Stage1 (reported)‡	44.3	35.6	21.7	23.1	31.2
OPSD-PRM (ours)	**52.94**	**43.28**	**30.93**	**34.39**	**40.38**

Table 22: Resources used to construct process-verification supervision. Human step labels concern candidate-step judgments, distinct from worked reference solutions.

Method / recipe	Supervision information	Construction resources
Qwen-Math-PRM800K	Human step judgments	PRM800K process annotations (Zheng et al., 2025; Lightman et al., 2024)
Math-Shepherd / MC-HE	Continuation-derived labels	Sampled completions; answer checking (Wang et al., 2024)
Eurus Stage1 / Implicit CE	Binary outcome labels	Candidate responses; answer checking (Yuan et al., 2025; PRIME-RL, 2025a)
Eurus Stage2	Outcomes and synthetic step labels	Error insertion by Llama-3.1-70B-Instruct and Qwen2.5-72B-Instruct (PRIME-RL, 2025b)
GenPRM	MC/RPE labels; verification rationales	Rollouts, QwQ-32B, consensus filtering (Zhao et al., 2026a)
ThinkPRM (main recipe)	Rationales filtered with human step labels	QwQ-32B-Preview; PRM800K annotations (Khalifa et al., 2026)
OPSD-PRM	Reference-conditioned evaluation distributions	Native references; same-backbone teacher

ThinkPRM separately evaluates a variant using MC-derived labels (Khalifa et al., 2026). Our references already accompany the training release, so no additional reference collection is performed. Online evaluation generation, teacher forward passes, and student optimization incur computation. Appendix B.1 describes the shared pre-generated candidates and their upstream source.

Table 23: Published and local verifier results. F1 (%); each row identifies its source. Stage2's model-card Macro-4 averages its four subset scores. Training resources and decoding follow the respective sources; GenPRM report rows use arXiv v1, Table 1.

Model / source	GSM8K	MATH	Olympiad	Omni	Macro-4
Eurus-Stage1 / GenPRM report	44.3	35.6	21.7	23.1	31.2
Eurus-Stage1 / local	54.65	41.22	24.70	17.53	34.52
Eurus-Stage2 / GenPRM report	47.3	35.7	21.2	20.9	31.3
Eurus-Stage2 / official card	67.0	53.2	35.4	30.7	46.58
GenPRM-7B / Pass@1	78.7	80.3	72.2	69.8	75.2

E Candidate Composition and Replication

We report main-model replication and the fixed-pool candidate-composition study.

E.1 Main-Model Replication and Initialization

Table 24 adds two seeds to the main 20260914 run; Table 25 compares the main model with its initialization under the same prompt, greedy decoding, and verdict readout.

Table 24: Main-model replication. ProcessBench F1 (%); SD is sample standard deviation across three training seeds.

Seed	GSM8K	MATH	Olympiad	Omni	Macro-4
20260914	52.94	43.28	30.93	34.39	40.38
20260919	50.30	40.52	32.00	36.37	39.80
20260920	56.37	42.32	32.66	33.89	41.31
Mean	53.21	42.04	31.86	34.88	40.50
Sample SD	3.04	1.40	0.87	1.31	0.76

Table 25: Initialization control under the same public prompt, greedy decoding, and verdict readout as the main student. F1 (%); bold marks the higher score per column.

Model	GSM8K	MATH	Olympiad	Omni	Macro-4
Qwen2.5-7B-Instruct	40.87	32.08	18.57	19.75	27.82
OPSD-PRM	**52.94**	**43.28**	**30.93**	**34.39**	**40.38**

E.2 Composition Protocol and Results

Each of 1,024 paired problems supplies one fixed successful or failed candidate, giving 128/896, 256/768, or 512/512 successful/failed counts with nested successful-problem sets. Both methods use two visits, global batch 32, 64 updates, and initial learning rate 5×10^{-6}. Mean token lengths are 385.79/385.75/385.99; mean step counts are 10.95/10.89/10.93. This pool differs from the natural-composition scaling pool.

Seed 20260914 covers all proportions; 20260915 repeats 12.5% and 50%, changing LoRA initialization and OPSD sampling while fixing problems, candidates, assignments, and visit order. Original 25% F1 is OPSD 51.09/43.91/32.79/34.37 (GSM8K/MATH/Olympiad/Omni), Test-3 37.02; CE 33.32/28.60/20.08/23.15, Test-3 23.95. There is no second-seed 25% run.

Table 26: Composition endpoint replications. F1 (%); the same 514 Test-3 core examples define both seeds' local rates.

Seed	Method	Success	GSM8K	MATH	Olympiad	Omni	Test-3	Core exact	Core accept
0914	OPSD	12.5%	48.61	41.23	32.64	35.28	36.38	23.54	30.54
0914	CE	12.5%	26.50	26.99	19.06	18.54	21.53	13.23	34.63
0914	OPSD	50%	46.18	40.64	28.50	29.04	32.73	20.43	43.19
0914	CE	50%	36.90	31.27	22.32	23.37	25.65	16.73	39.11
0915	OPSD	12.5%	48.71	42.12	35.13	34.44	37.23	26.26	31.91
0915	CE	12.5%	26.50	27.68	20.40	17.71	21.93	13.62	34.24
0915	OPSD	50%	47.18	41.16	29.00	25.79	31.98	18.87	46.30
0915	CE	50%	38.06	31.02	20.60	24.33	25.32	18.09	35.99

Table 27: Interaction $D = (\mathrm{OPSD} - \mathrm{CE})_{12.5\%} - (\mathrm{OPSD} - \mathrm{CE})_{50\%}$, in pp. Intervals resample evaluation problems and are calculated separately for each seed.

Training seed	Test-3 F1	Core exact	Core false acceptance
20260914	$+7.78\ [4.56, 10.89]$	$+6.61\ [1.35, 11.85]$	$-8.17\ [-13.86, -2.60]$
20260915	$+8.63\ [5.47, 11.71]$	$+11.87\ [6.36, 17.53]$	$-12.65\ [-18.55, -7.22]$

Table 26 abbreviates seeds to 0914/0915. GSM8K-selected CE thresholds at 12.5%/50% are 0.3656828414/0.4847423712 for 20260914 and 0.3641820910/0.4862431216 for 20260915; the 25% threshold is 0.4217108554. Core results use 514 Test-3 examples, excluding GSM8K's seven. Replication varies training randomness with the candidate pool fixed.