REWARDING REASONING, NOT ANSWERS: FIXING AND BOUNDING TEST-TIME REINFORCEMENT LEARNING ON MEDICAL QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time reinforcement learning adapts a model on its own unlabeled test set using majority-vote pseudo-labels and has shown strong results in mathematics. We show that this recipe collapses on medical multiple-choice QA: accuracy stagnates while output diversity rapidly declines. Through a controlled experiment that keeps the questions, model, and optimizer fixed while changing only the answer space, we trace this failure to answer-space structure rather than domain difficulty. In small answer spaces, incorrect rollouts often collide on the same wrong pseudo-label and reinforce it; in large answer spaces, they disperse and receive little reward. This diagnosis motivates PROSE, Process-Reward-Guided Self-Training, which rewards reasoning quality instead of answer agreement. PROSE scores each reasoning step with a medical process reward model, assigns the trajectory reward as the minimum score across steps, and enforces answer-format constraints. Without labels, PROSE substantially improves a general Llama model, surpassing purpose-built medical models and matching much larger systems. Because the process signal is internalized into the policy, the adapted model requires no reward model at inference and transfers its gains to unseen datasets. We further show that the minimum aggregation is essential: mean aggregation can be exploited, saturating the proxy reward while degrading accuracy.

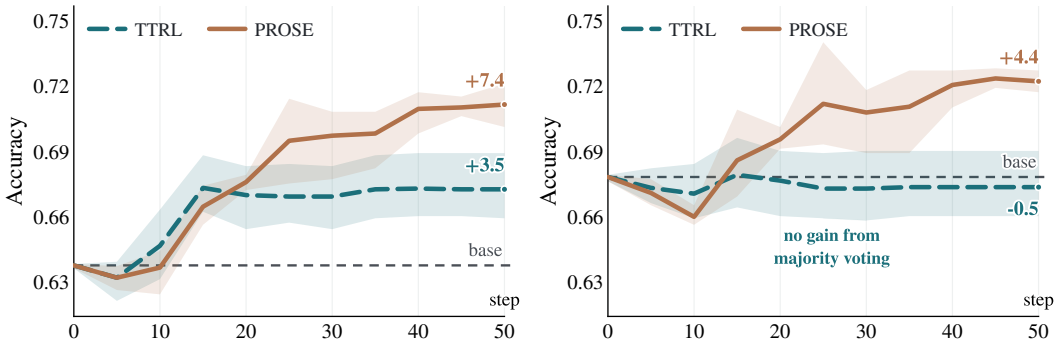


Figure 1: **Rewarding reasoning prevents test-time RL collapse.** Standard TTRL yields only modest gains in average accuracy over 16 sampled responses (avg@16) **(a)** and no gain in majority-vote accuracy over 16 sampled responses (maj@16) **(b)**, whereas PROSE consistently improves both metrics. Curves show the mean, and shaded regions denote $\pm$ one standard deviation across three random seeds.

1 INTRODUCTION

Recent work shows that large language models can adapt at test time using their own sampled outputs (Zuo et al., 2026; Moradi et al., 2025; Yang et al., 2026). Test-time reinforcement learning

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.

(TTRL) turns these outputs into a learning signal: for each unlabeled test question, the model samples multiple responses, treats the majority answer as a pseudo-label, and updates itself to agree with it. This simple recipe has produced strong gains in mathematics, making it attractive since it requires only test questions rather than ground-truth labels or additional annotation.

Medical question answering is a natural setting where such label-free adaptation would be especially valuable. Expert annotation is costly, clinical distributions vary across institutions and specialties, and privacy constraints often limit external labeling (Wang et al., 2021; Zech et al., 2018; Price & Cohen, 2019). At the same time, many medical benchmarks and decision-support interfaces are formulated as multiple-choice QA (MCQ), where the answer space is small and highly structured. This format raises a central question: does majority-vote self-training remain reliable when many different reasoning paths must choose from the same limited set of answers?

Figure 1 illustrates the failure that motivates this work: standard TTRL yields marginal gains in sample-level accuracy and none in majority-vote accuracy, while output diversity drops sharply. PROSE, which replaces answer agreement with a process-level reward, improves both metrics, indicating that the limitation lies in the reward signal rather than in test-time adaptation itself.

To isolate the source of this collapse, we conduct a controlled experiment that holds the questions, model, optimizer, and training procedure fixed, and varies only the answer space. The results support an answer-space explanation rather than a domain-difficulty explanation. We show that the *Lucky Hit* mechanism identified by TTRL (Zuo et al., 2026) depends on the structure of the answer space: in a large free-form answer space, incorrect rollouts usually disperse across distinct strings and are rarely reinforced by the majority-vote reward; in a small multiple-choice answer space, different mistaken rationales can converge on the same wrong option, producing a false consensus that is then rewarded as a pseudo-label. We quantify this effect through the protection gap between pseudo-label reward accuracy and true answer accuracy, and find that the gap present under free-form answering nearly vanishes under multiple choice.

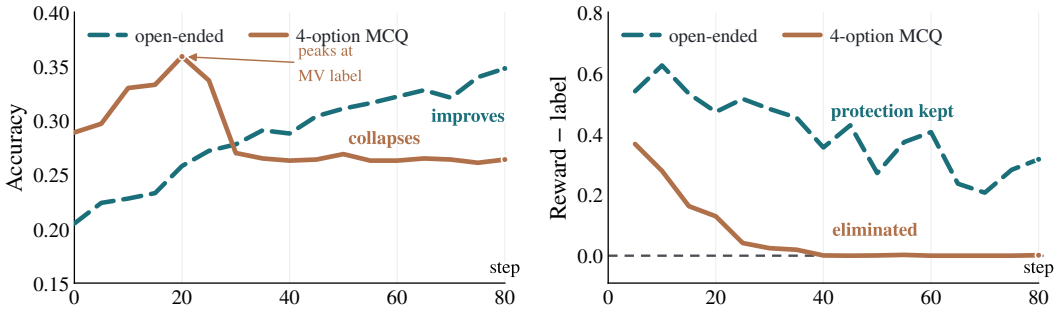
This diagnosis points to a different training signal. If agreement on the final answer is unreliable, the reward should evaluate the reasoning process that produced the answer (Uesato et al., 2022; Lightman et al., 2023). We introduce PROSE, which keeps the test-time RL loop but replaces majority-vote pseudo-labels with a step-level medical process reward model. PROSE scores each reasoning step, sets the trajectory reward to the minimum step score, and applies an answer-format guard. Minimum aggregation makes the reward depend on the weakest step in the chain, preventing trajectories from hiding flawed reasoning behind many easy or fluent steps.

Beyond improving adaptation, PROSE has a practical deployment advantage: the PRM is used only during adaptation, and the adapted policy is evaluated on its own with ordinary sampling or majority voting. The process signal is thus absorbed into the model parameters rather than applied as an external selector at inference time. PROSE is label-free with respect to the target set: no ground-truth answers are used for training. It does rely on a frozen, off-the-shelf process reward model, so the fair comparisons are not only against TTRL but against the same PRM used in two other ways: as an inference-time selector at a matched sampling budget (Section 6.1), and as a source of hard pseudo-labels within the same RL loop (Section 6.2). To sum up, our contributions are:

- We provide a controlled diagnosis of why TTRL fails on medical multiple-choice QA. The Lucky Hit mechanism shows how small answer spaces convert independent reasoning errors into reinforced pseudo-label errors.
- We introduce PROSE, a label-free test-time adaptation method that rewards reasoning quality with a medical process reward model, assigns trajectory rewards by the minimum score across reasoning steps, and guards the answer format.
- We show that PROSE internalizes the process reward into the policy, requires no reward model at inference, generalizes beyond the adapted datasets, and avoids the reward hacking observed with mean aggregation.

2 Related Work

TTS and Self-Evolution. Increasing computation at inference time has become a practical way to elicit stronger reasoning from LLMs, often rivaling gains from scaling model



(a) Avg@16 (b) Protection gap

Figure 2: Changing only the answer space flips TTRL from improvement to collapse. **(a)** Avg@16 improves for open-ended answers but collapses for 4-option MCQs. **(b)** The protection gap (reward accuracy − label accuracy) remains positive for open-ended answers but vanishes for 4-option MCQs.

size (Snell et al., 2024). A dominant line of work samples multiple solutions and aggregates them via consensus (e.g., Self-Consistency) or selection (e.g., best-of-N with a reward model) (Wang et al., 2022; Snell et al., 2024; Huang et al., 2025). Beyond selection-only schemes, TTRL enables parameter adaptation on unlabeled inputs by deriving proxy rewards from repeated sampling and MV, yielding *self-evolution* at test time (Zuo et al., 2026; Liu et al., 2025a; Pan et al., 2026); unlike RL with verifiable rewards or human annotation (Guo et al., 2025; Jaech et al., 2024), this is appealing in medical settings where gold standards are rarely available in real time (Hager et al., 2024).

Inference-Time Verification: From Outcome to Process Rewards. Test-time scaling improves output quality without updating model parameters, typically by exploring multiple reasoning trajectories and selecting the most reliable one. Outcome-based reward models score whole solutions but can misjudge trajectories due to spurious reasoning that happens to reach the correct answer or correct early steps that later derail (Yuan et al., 2025; Lee et al., 2025). Process reward models address this by providing stepwise supervision to evaluate intermediate reasoning validity (Lightman et al., 2023; Zheng et al., 2025). In medicine, stepwise verification is challenging due to the lack of symbolic verifiers; Med-PRM tackles this by grounding step evaluations in retrieved clinical guidelines and medical literature (RAG-as-a-judge), enabling stronger verifier-based selection in medical reasoning (Yun et al., 2025; Liu et al., 2025b). Step-level rewards have also been used to train policies with labeled training sets, e.g., Math-Shepherd, PRIME, and process-advantage verifiers (Wang et al., 2023; Cui et al., 2025a; Setlur et al., 2024); PROSE instead applies a process reward transductively on the unlabeled test set, after first establishing why outcome-level consensus fails there.

3 Motivation: TTRL Fails on Medical MCQ

Observed Collapse. We first examine standard majority-vote TTRL on medical multiple-choice QA. As shown in Figure 1, TTRL produces only limited gains in sample-level accuracy, while majority-vote accuracy remains essentially unchanged. At the same time, output diversity drops sharply (Cui et al., 2025b; Yue et al., 2026). The model therefore becomes more self-consistent without becoming more correct: TTRL rewards agreement with the majority answer, but in medical MCQ agreement is not a reliable proxy for correctness.

Answer Space Matters. One possible explanation is that medical questions are simply harder than the mathematics tasks where TTRL has succeeded. To test this, we construct a controlled experiment on AMC mathematics problems (Zuo et al., 2026). Each question is evaluated in two formats: its original free-form format and a matched four-option multiple-choice format. The questions, policy model, optimizer, and training procedure are held fixed; only the answer format changes, from free-form to four-option multiple choice.

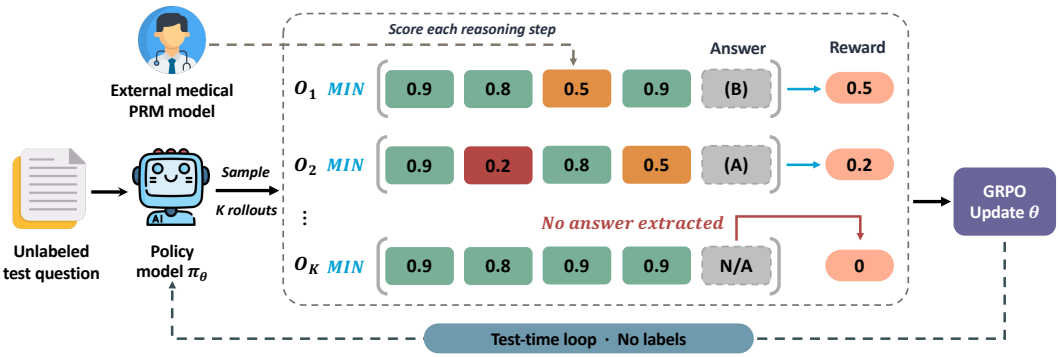


Figure 3: **Overview of PROSE.** Given an unlabeled test question, the policy π_θ samples K rollouts. An external medical process reward model scores each reasoning step, and the minimum step score defines the reward for each rollout. A format guard assigns zero reward when no answer can be extracted. GRPO then updates the policy using the rollouts and their rewards, and the process repeats on the test set without labels.

Figure 2 shows that this change is sufficient to reverse the training dynamics. In the free-form setting, TTRL steadily improves accuracy. In the matched multiple-choice setting, accuracy initially rises but later collapses. Since the underlying questions and optimization procedure are unchanged, the result points to answer-space structure rather than domain difficulty as the primary driver of failure.

The Lucky Hit Mechanism. The controlled experiment reveals how the *Lucky Hit* effect depends on the answer space. In a large free-form answer space, incorrect rollouts are dispersed across many distinct strings. Even when the majority pseudo-label is wrong, most other wrong rollouts do not exactly match it and therefore receive no reward. This sparsity protects training from many pseudo label errors. In a small multiple-choice answer space, this protection disappears. Different mistaken rationales are compressed into the same few options, so wrong rollouts often collide on the same incorrect answer. Once that answer becomes the majority pseudo-label, many flawed trajectories receive positive reward. We quantify this protection as the *protection gap*: the fraction of rollouts whose majority-vote reward agrees with the ground-truth reward, minus the fraction of questions whose pseudo-label is correct (Appendix B). The measured gap falls from strongly positive in the free-form setting to nearly zero under multiple choice. This mechanism also explains why diversity collapse alone is not the root cause. Diversity can decrease while accuracy improves, as in the free-form condition. What matters is where the probability mass concentrates. In medical MCQ, the small answer space allows probability mass to concentrate on a wrong option, turning a pseudo-label error into a self-reinforcing training signal.

4 METHOD

The preceding diagnosis shows that majority-vote TTRL fails because final-answer agreement becomes unreliable in a small answer space. PROSE (**Pro**cess-Reward-Guided **Se**lf-Training) keeps the same test-time RL loop but replaces answer-level pseudo-label rewards with process-level rewards. Figure 3 gives an overview; the entire procedure uses only unlabeled test questions.

Test-Time RL Setup. We consider transductive test-time adaptation on an unlabeled test set $\mathcal{D} = \{q\}$. For each question q, the policy samples a group of K rollouts,

$$\{o_k\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid q), \tag{1}$$

where each rollout $o_k = (s_1^{(k)}, \ldots, s_{n_k}^{(k)}, a_k)$ contains a sequence of reasoning steps followed by a final answer a_k.

Given rollout rewards $\{r_k\}_{k=1}^{K}$, we optimize the policy with GRPO (Shao et al., 2024). The advantage of rollout k is normalized within the sampled group,

$$A_k = \frac{r_k - \mu}{\sigma}, \tag{2}$$

where μ and σ are the mean and standard deviation of the group rewards. GRPO then updates θ with the clipped policy-gradient objective.

Standard TTRL defines reward by matching the majority-vote answer,

$$r_{k,\mathrm{TTRL}} = \begin{cases} 1, & \text{if } a_k = \hat{a}, \\ 0, & \text{otherwise}, \end{cases} \qquad \hat{a} = \mathrm{MV}\big(\{a_j\}_{j=1}^{K}\big). \tag{3}$$

Process-Level Reward. PROSE evaluates each rollout by the quality of its reasoning process rather than by agreement with other sampled answers. A medical PRM assigns a correctness score to each reasoning step,

$$p_i^{(k)} = \mathrm{PRM}\Big(s_i^{(k)} \mid q, s_{<i}^{(k)}\Big) \in [0,1]. \tag{4}$$

This score is independent of whether other rollouts choose the same final option, so it is not directly affected by answer-space collisions.

To convert step scores into a trajectory reward, PROSE uses the minimum score across reasoning steps,

$$r_k = \min_i p_i^{(k)}. \tag{5}$$

This aggregation matches the structure of multi-step reasoning: a chain is only as reliable as its weakest step. It also limits reward hacking (Gao et al., 2023; Skalse et al., 2022; Amodei et al., 2016). Mean or sum aggregation can be increased by adding high-scoring steps that dilute a flawed one, whereas the minimum can improve only when the weakest step improves. Empirically, this distinction is important: mean aggregation saturates the proxy reward while degrading accuracy, whereas minimum aggregation remains aligned with answer correctness.

Format Guard. The PRM scores reasoning steps but does not evaluate whether the rollout ends with a valid final answer. Without an explicit constraint, optimization can favor high-scoring reasoning traces that fail to produce a parseable answer. PROSE therefore applies a format guard:

$$r_k = \begin{cases} \min_i p_i^{(k)}, & a_k \text{ is parseable}, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

A rollout receives high reward only when it contains both a valid final answer and consistently reliable reasoning steps.

Algorithm and Deployment. Algorithm 1 in Appendix C summarizes one PROSE update. The PRM provides rewards only during test-time adaptation; the adapted policy is then evaluated with ordinary sampling or majority voting, so PROSE uses the PRM as a temporary training signal rather than as a per-query selector.

5 Experimental Setup

Policies. We evaluate two model families and three sizes: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-$\{$1.7B, 4B, 8B$\}$ (with thinking disabled) (Yang et al., 2025); the two 8B models are our main policies and the smaller Qwen3 sizes serve the model-scale study.

Process reward model. The reward comes from Med-PRM (Yun et al., 2025), an 8B medical process reward model that assigns each reasoning step a probability of being correct, $p_i(+) \in [0,1]$, using the retrieval pipeline released with Med-PRM. Med-PRM is used only during test-time training: it provides the per-step scores that PROSE aggregates into a reward; the adapted policy answers at inference without it.

Datasets and protocol. We use four clinical multiple-choice benchmarks: MedQA-5op (5-option), MedQA-4op (4-option) (Jin et al., 2021), MedMCQA (Pal et al., 2022), and DDX-Plus (Fansi Tchango et al., 2022). Following the test-time RL setting, each run is transductive: the training set is the test set and no ground-truth labels are used for training. A separate run adapts each policy on each dataset. All evaluation questions are drawn from the held-out test splits used to evaluate Med-PRM and were never used in its training; MedQA-5op and MedQA-4op are disjoint subsets of the MedQA test set with no shared questions (Appendix A).

Table 1: Avg@16 performance on four medical QA datasets. Our method consistently improves standard TTRL across all datasets. Best and second-best results among models with fewer than 32B parameters are shown in **bold** and underlined, respectively.

Category	Model	Size	MedQA-5op	MedMCQA	DDXPlus	MedQA-4op	Avg.
Large-scale General-purpose LLMs	Gemini Flash 2.0	–	0.852	0.726	0.750	0.875	0.801
	GPT-4o-mini	–	0.743	0.682	0.760	0.790	0.744
	GPT-3.5 turbo	–	0.654	0.570	0.738	0.699	0.665
	QWQ	32B	0.769	0.609	0.761	0.825	0.741
	R1-Distill-Qwen	32B	0.738	0.591	0.781	0.772	0.721
	Sky-T1	32B	0.749	0.587	0.776	0.795	0.727
Open-source Small Language Models	Llama-3.1	8B	0.636	0.556	0.695	0.654	0.635
	Gemma2	9B	0.532	0.512	0.676	0.581	0.575
	Ministral	8B	0.475	0.469	0.606	0.530	0.520
Open-source Medical Language Models	Med42	8B	0.575	0.573	0.597	0.614	0.590
	Meditron3	8B	0.569	0.549	0.632	0.612	0.591
	OpenBioLLM	8B	0.363	0.459	0.438	0.373	0.408
	m1-7B	7B	<u>0.681</u>	0.568	0.662	<u>0.721</u>	0.658
	Meerkat	8B	0.654	0.569	0.709	0.672	0.651
	UltraMedical	8B	0.665	0.579	0.694	0.710	0.662
	HuatuoGPT-o1	8B	0.666	<u>0.619</u>	0.660	0.714	0.665
Test-time Training on Llama-3.1-8B	TTRL	8B	0.672	0.573	<u>0.761</u>	0.676	<u>0.671</u>
	Ours	8B	**0.724**	**0.623**	**0.858**	**0.756**	**0.740**

Baselines. We compare against: (i) the base model with no adaptation; (ii) TTRL, test-time RL with the majority-vote reward (Zuo et al., 2026); (iii) inference-time Med-PRM selection at a matched sampling budget, including best-of-N reranking (BoN) and reward-weighted self-consistency (SC+RM); and (iv) Large-scale General-purpose LLMs, open-source small language models and medical models re-run under our protocol (Chen et al., 2024; Zhang et al., 2024; Team et al., 2024).

Metrics. Our primary metric is avg@16: the average accuracy over 16 samples per question. Where we compare against inference-time selection (BoN, SC+RM), whose output is a single chosen answer, we instead report maj@16, the accuracy of the majority vote over 16 samples, so that all methods spend the same sampling budget. All models, including the external baselines, are evaluated under the same sampling protocol (Appendix C.6). Main tables report a single seed (1000). Across three seeds on MedQA-5op (Figure 1, Appendix C.7), the standard deviation of avg@16 is 0.004 for PROSE and 0.014 for TTRL, well below the 5–10 point gaps between PROSE and the baselines.

Optimization. We optimize with GRPO. Each update step uses 7 prompts $\times$ 32 samples (224 rollouts), and we train for 10 epochs with AdamW (learning rate 5×10^{-7}), a symmetric PPO clip of 0.2, and a KL penalty with coefficient 10^{-3}. Training rollouts use temperature 1.0. The full configuration is given in Table 5 of Appendix C.

6 Results and Analysis

6.1 Main Results

PROSE lifts an 8B model to the top of its class. Table 1 reports avg@16 across the four clinical datasets, with every model evaluated under the same protocol. Applied to Llama-3.1-8B, PROSE reaches an average of 0.740, the best among all 8–9B general and medical models: 10.5 points above its base model (0.635) and 6.9 points above majority-vote TTRL on the same backbone (0.671), so the gain comes from the process reward rather than from test-time training alone. It also surpasses every purpose-built open-source medical model, the strongest of which reaches 0.665, despite using a general model adapted without labels.

Table 2: Comparison with TTRL and PRM-based inference methods on four medical QA datasets. PROSE achieves the best average performance for both model families and the best result in seven of eight model-dataset settings. Best results are shown in **bold**.

Model	Test-Time Method	MedQA-5op	MedMCQA	DDXPlus	MedQA-4op	Avg.
Llama-3.1-8B	TTRL (maj@16)	0.684	0.572	0.760	0.714	0.683
	PRM (BON@16)	0.690	0.610	0.780	0.720	0.700
	PRM (SC+RM@16)	0.690	0.620	0.740	0.740	0.698
	PROSE (maj@16)	**0.740**	**0.628**	**0.863**	**0.764**	**0.749**
Qwen3-8B	TTRL (maj@16)	0.696	0.606	0.866	0.729	0.724
	PRM (BON@16)	0.690	0.590	0.810	**0.770**	0.715
	PRM (SC+RM@16)	0.720	0.580	0.840	0.760	0.725
	PROSE (maj@16)	**0.754**	**0.645**	**0.907**	0.760	**0.767**

The adapted 8B model is also competitive with much larger systems: its 0.740 average is within 0.004 of QwQ-32B (0.741), with roughly one quarter of the parameters, and GPT-4o-mini (0.744), with only Gemini Flash 2.0 (0.801) clearly ahead. Among models of comparable size it performs best on all four datasets, and on DDXPlus it reaches 0.858, the highest score in the entire table, surpassing every proprietary and 32B model, while proprietary and 32B models retain an advantage on the exam-style MedQA and MedMCQA datasets, though PROSE closes much of this gap with a small, open model and no labels.

Reshaping the policy beats reranking its samples. Med-PRM is designed for inference-time selection with BoN or SC+RM. We ask whether the same reward model is more effective when used for selection or test-time training. Since BoN and SC+RM return a single answer, we evaluate PROSE using maj@16 from the same $N{=}16$ samples, ensuring a comparable single-answer evaluation and inference-time sampling budget. Table 2 compares these approaches on two 8B policies. The comparison with TTRL directly isolates the effect of the reward signal, as the two methods otherwise follow the same test-time RL framework. PROSE improves the average by 6.6 percentage points on Llama-3.1-8B ($0.683 \rightarrow 0.749$) and by 4.3 points on Qwen3-8B ($0.724 \rightarrow 0.767$), outperforming TTRL on every dataset. PROSE also substantially outperforms inference-time PRM selection, achieving the best result in 7 of the 8 dataset$\times$model settings; the only exception is MedQA-4op on Qwen3-8B, where best-of-N leads by 0.010. Compared with the strongest PRM selection baseline, it improves the average from 0.700 to 0.749 on Llama-3.1-8B and from 0.725 to 0.767 on Qwen3-8B. The largest gains occur on DDXPlus, where PROSE reaches 0.863 with Llama-3.1-8B and 0.907 with Qwen3-8B.

These results speak to how the PRM is best used. BoN and SC+RM can only pick among samples drawn from a fixed model, whereas PROSE uses the same Med-PRM to reshape the policy's output distribution, making correct reasoning more likely in the first place; at a matched sampling budget, reshaping beats reranking. The reward is thereby *internalized* into the weights rather than merely consulted: the adapted policy answers by plain majority voting with no PRM at inference, while BoN and SC+RM query the PRM on every question. We do not claim a net compute saving, since test-time training is itself expensive; rather, the PRM is needed only during the adaptation phase, so its cost amortizes whenever the adapted policy is reused. We test this internalization next through out-of-distribution generalization to datasets never used for adaptation.

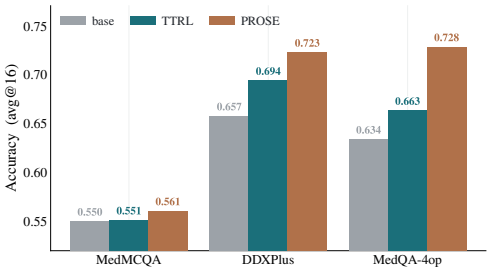
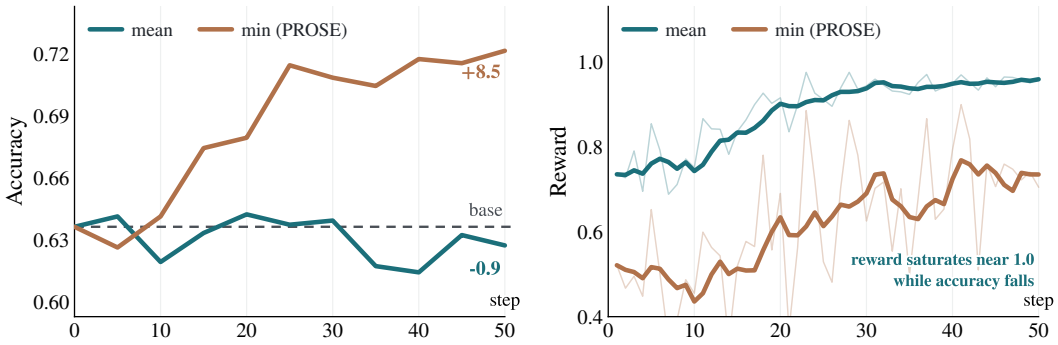


Figure 4: Out-of-distribution generalization (avg@16). The policy adapted on MedQA-5op is frozen and evaluated on three unseen datasets without further adaptation or PRM scoring; PROSE outperforms both the base model and TTRL.

Out-of-distribution generalization. If PROSE merely fit the specific questions used for adaptation, its gains would not transfer across datasets. We therefore freeze the policy adapted on MedQA-



(a) Avg@16 (b) PRM reward (train)

Figure 5: MIN vs. mean aggregation of step-level PRM scores on MedQA-5op. **(a)** With MIN aggregation (PROSE), accuracy improves and remains above the untrained base, whereas mean aggregation eventually falls below it. **(b)** Despite the accuracy degradation, the mean-aggregated PRM reward saturates near 1.0, indicating reward hacking. Faint lines show the raw per-step rewards, and thick lines show their 5-step moving averages.

5op and evaluate it on three datasets unseen during adaptation, whose questions are disjoint from the adaptation set, using standard sampling without the PRM. Figure 4 shows that PROSE improves over the base model on all three out-of-distribution datasets and outperforms TTRL in every case: MedMCQA improves from 0.550 to 0.561, DDXPlus from 0.657 to 0.723, and MedQA-4op from 0.634 to 0.728. As expected, these transferred gains are smaller than those obtained by adapting directly to each target dataset (see Table 1). The gain is largest on MedQA-4op, which most closely resembles the adaptation set, and smallest on MedMCQA.

This is the clearest evidence that the process reward has been internalized. The adapted policy carries its improvement to new distributions without any PRM at inference. The external evaluator was needed only during adaptation, to bootstrap the policy, and what remains is a self-contained model rather than a set of memorized answers.

6.2 ABLATION: WHAT MATTERS IN THE REWARD DESIGN

A process reward beats a process label. Table 3 isolates the first choice on MedQA-5op. Replacing majority voting with a PRM-derived pseudo-label, namely the final answer of the rollout with the highest min-aggregated PRM score used as a hard 0/1 target in the same RL loop, already improves avg@16 from 0.672 to 0.709, confirming that the process signal captures information missed by majority voting. But keeping the PRM as a continuous reward is better still (0.724 avg@16), and the gap widens under maj@16. The reason is what each formulation discards: collapsing the step scores into a single target to imitate throws away their gradation, so every trajectory off the chosen pseudo-label gets zero reward regardless of how close it came, whereas a graded reward preserves the ordering among partially-correct trajectories and tells the policy which of its attempts came closer.

Table 3: Effect of incorporating process supervision into TTRL on MedQA-5op with Llama-3.1-8B. PRM-derived labels outperform majority-vote labels, while using PRM scores as graded rewards (PROSE) achieves the best avg@16 and maj@16 performance. Best results are shown in **bold**.

Method	avg@16	maj@16
MV as label	0.672	0.684
PRM as label	0.709	0.710
PRM as reward (PROSE)	**0.724**	**0.740**

MIN aggregation keeps the reward honest. The second design choice is how to aggregate the step-level PRM scores, which directly determines the method's susceptibility to reward hacking. Figure 5 compares minimum and mean aggregation under otherwise identical settings. With minimum aggregation, avg@16 improves by 8.5 percentage points over the untrained base and remains consistently above it. With mean aggregation, avg@16 eventually falls 0.9 points below the base, even as the optimized PRM reward approaches 1.0. The proxy is therefore nearly saturated while the

Table 4: Effect of model scale on single-answer accuracy at a budget of 16 samples. PROSE performs best across all four datasets at 4B, while at 1.7B its advantage is limited to DDXPlus, indicating that effective process-reward training depends on sufficient policy capacity. Best results within each model block are shown in **bold**.

Model	Test-Time Method	MedQA-5op	MedMCQA	DDXPlus	MedQA-4op	Avg.
Qwen3-4B	TTRL (maj@16)	0.626	0.512	0.814	0.657	0.652
	PRM (BON@16)	0.680	0.570	0.790	0.700	0.685
	PRM (SC+RM@16)	0.670	0.570	0.800	0.690	0.683
	PROSE (maj@16)	**0.725**	**0.621**	**0.856**	**0.733**	**0.734**
Qwen3-1.7B	TTRL (maj@16)	0.436	0.462	0.826	0.520	0.561
	PRM (BON@16)	**0.570**	**0.570**	0.800	**0.590**	**0.633**
	PRM (SC+RM@16)	0.530	0.510	0.810	0.570	0.605
	PROSE (maj@16)	0.493	0.553	**0.871**	0.540	0.614

true objective deteriorates, a clear sign of reward hacking, as anticipated in Section 4: mean aggregation allows low-scoring reasoning steps to be diluted by additional high-scoring steps, enabling the policy to increase its reward without correcting its weakest reasoning. Minimum aggregation blocks this strategy because the trajectory reward cannot exceed its lowest step-level score, so optimization must improve the weakest step; accordingly, the minimum-aggregated reward stays well below saturation (0.73–0.75) while accuracy improves.

6.3 Ablation: Effect of model scale

Table 4 asks whether internalizing the reward still pays off for smaller policies, comparing PROSE with TTRL and inference-time PRM selection on Qwen3-4B and Qwen3-1.7B at the same budget of $N{=}16$. At 4B the picture is unchanged from 8B: PROSE is the best method on all four datasets and by a wide margin on average (0.734 vs. 0.685 for the strongest selector), so the benefit of training with the process reward is not an artifact of scale. At 1.7B, however, the ordering reverses on three of four datasets—inference-time best-of-N leads on average (0.633 vs. 0.614)—and PROSE prevails only on DDXPlus.

This reversal is informative rather than incidental. Training with the process reward requires the policy to be able to produce better reasoning once it is pushed toward it, whereas selection only requires that a good trajectory occasionally appear among N samples and be recognizable to the PRM. For a very weak policy the second condition is easier to satisfy, so internalization pays off only once the policy is strong enough to act on the signal, from roughly 4B in our setting.

7 Conclusion

We showed that test-time RL, effective on mathematics, collapses on medical multiple-choice QA for a structural reason: in a small answer space, incorrect rollouts collide on the same wrong pseudo-label and reinforce it. A controlled experiment varying only the answer format supports this explanation, and the reward−label protection gap quantifies it. PROSE follows from this diagnosis, rewarding the reasoning process rather than answer agreement through minimum step-level aggregation and an answer-format guard. Without labels, it gives Llama-3.1-8B the best average accuracy among the evaluated small models and matches much larger systems. The adapted policy requires no reward model at inference, outperforms PRM selection at the same sampling budget, and retains its gains on datasets unseen during adaptation.

Limitations and future work. PROSE requires a domain-relevant process reward model. Self-generated or general-purpose process rewards could reduce this dependence and extend it beyond medical QA. Its effectiveness also depends on policy capacity: inference-time selection remains preferable below roughly 4B parameters in our experiments. Future work should characterize this transition and how policy capacity and reward quality affect reward internalization.

AI USE STATEMENT

In this work, we used generative AI tools only to aid and polish the writing of the manuscript, that is, to improve grammar, clarity, and readability of text drafted by the authors, and to assist with LaTeX formatting. We have not used generative AI tools for any task requiring disclosure under the ICLR 2027 AI Policy for Authors: generating synthetic data, developing the theoretical framing or hypotheses, designing the methodology or experiments, implementing methods, writing proofs, cleaning or reformatting datasets, analyzing data, or interpreting results; translation is not applicable to this work. All AI-assisted edits were reviewed by the authors for accuracy and fidelity to the intended meaning. We take responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work uses only publicly available medical question-answering benchmarks (MedQA, MedMCQA, DDXPlus) and public mathematics problems (AMC). DDXPlus consists of synthetically generated patient cases and contains no real patient records; the remaining datasets are derived from licensing and entrance examinations and contain no protected health information. No human subjects were involved, and no institutional review was required. The models studied answer examination-style questions and are not validated for clinical use; they should not be used for diagnosis or treatment decisions. Test-time adaptation can amplify existing model errors when the reward signal is unreliable, as our diagnosis shows, and practitioners should validate adapted models on labeled data before any deployment. The authors declare no competing interests.

REPRODUCIBILITY STATEMENT

We have taken several steps to make our results reproducible. The method is fully specified in Section 4 and Algorithm 1, and all training hyperparameters are listed in Section 5 and Table 5 of Appendix C. Appendix A describes every dataset, including the exact construction of the matched multiple-choice AMC variant used in the controlled experiment. Appendix B gives formal definitions of all reported metrics, and Appendix C.4 reproduces the exact prompts given to the policy and to the process reward model. Random seeds and seed-level variance are reported in Appendix C. All base models (Llama-3.1-8B-Instruct, Qwen3, Med-PRM) and datasets are publicly available. Source code and adaptation scripts will be released upon publication.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. HuatuoGPT-o1, towards medical complex reasoning with LLMs. *arXiv preprint arXiv:2412.18925*, 2024.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025a.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025b.

Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. DDXPlus: A new dataset for automatic medical diagnosis. *Advances in neural information processing systems*, 35:31306–31318, 2022.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

P. Hager, F. Jungmann, R. Holland, K. Bhagat, I. Hubrecht, M. Knauer, J. Vielhauer, M. Makowski, R. Braren, G. Kaissis, and D. Rueckert. Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30:2613 – 2622, 2024. doi: 10.1038/s41591-024-03097-1.

Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. m1: Unleash the potential of test-time scaling for medical reasoning with large language models. *arXiv preprint arXiv:2504.00869*, 2025.

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Dong Bok Lee, Seanie Lee, Sangwoo Park, Minki Kang, Jinheon Baek, Dongki Kim, Dominik Wagner, Jiongdao Jin, Heejun Lee, Tobias Bocklet, et al. Rethinking reward models for multi-domain test-time scaling. *arXiv preprint arXiv:2510.00492*, 2025.

H. Lightman, Vineet Kosaraju, Yura Burda, et al. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, and ShaoGuo Liu. ETTRL: Balancing exploration and exploitation in LLM test-time reinforcement learning via entropy mechanism. *arXiv preprint arXiv:2508.11356*, 2025a.

Siru Liu, Allison B. McCoy, and Adam Wright. Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines. *Journal of the American Medical Informatics Association : JAMIA*, 32: 605 – 615, 2025b. doi: 10.1093/jamia/ocaf008.

Mohammad Mahdi Moradi, Hossam Amer, Sudhir Mudur, Weiwei Zhang, Yang Liu, and Walid Ahmed. Continuous self-improvement of large language models by test-time training with verifier-driven sample selection. *arXiv preprint arXiv:2505.19475*, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL `https://proceedings.mlr.press/v174/pal22a.html`.

Teng Pan, Yuchen Yan, Zixuan Wang, Ruiqing Zhang, Guiyang Hou, Wenqi Zhang, Weiming Lu, Jun Xiao, and Yongliang Shen. CoVerRL: Breaking the consensus trap in label-free reasoning via generator-verifier co-evolution. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29833–29853, 2026.

W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25 (1):37–43, 2019.

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. *arXiv preprint arXiv:2410.08146*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

C. Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Shanshan Wang, Cheng Li, Rongpin Wang, Zaiyi Liu, Meiyun Wang, Hongna Tan, Yaping Wu, Xinfeng Liu, Hui Sun, Rui Yang, et al. Annotation-efficient deep learning for automatic medical image segmentation. *Nature communications*, 12(1):5915, 2021.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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

Chengyi Yang, Zhishang Xiang, Yunbo Tang, Zongpei Teng, Chengsong Huang, Fei Long, Yuhan Liu, and Jinsong Su. TTCS: Test-time curriculum synthesis for self-evolving. *arXiv preprint arXiv:2601.22628*, 2026.

Youliang Yuan, Qiuyang Mang, et al. Curing miracle steps in LLM mathematical reasoning with rubric rewards. *arXiv preprint arXiv:2510.07774*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Jaehoon Yun, Jiwoong Sohn, Jungwoo Park, Hyunjae Kim, Xiangru Tang, Yanjun Shao, Yonghoe Koo, Minhyeok Ko, Qingyu Chen, Mark Gerstein, Michael Moor, and Jaewoo Kang. Med-PRM: Medical reasoning models with stepwise, guideline-verified process rewards. *arXiv preprint arXiv:2506.11474*, 2025.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. UltraMedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems*, 37:26045–26081, 2024.

Congmin Zheng, Jiachen Zhu, Zhuoying Ou, et al. A survey of process reward models: From outcome signals to process supervisions for large language models. *arXiv preprint arXiv:2510.08049*, 2025.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Dataset Details

A.1 Medical Datasets

MedQA-5op. MedQA is a medical question-answering benchmark collected from professional medical licensing examinations in the United States, mainland China, and Taiwan (Jin et al., 2021). We use its English USMLE-style subset, which evaluates professional medical knowledge and clinical decision making through board-examination questions. Each question contains five candidate answers, labeled A–E, with exactly one correct option.

MedQA-4op. MedQA-4op is the official four-option version of the English MedQA benchmark, in which each question is restricted to four candidate answers labeled A–D. It preserves the medical domain and examination style of MedQA-5op while providing a smaller answer space. In our experiments, MedQA-5op and MedQA-4op are two disjoint subsets of the MedQA test set: no question appears in both, so adaptation on one and evaluation on the other (Section 6.1) involves no shared items.

MedMCQA. MedMCQA is constructed from questions appearing in the Indian AIIMS and NEET-PG medical entrance examinations (Pal et al., 2022). It covers a broad range of medical subjects, including anatomy, pathology, pharmacology, surgery, internal medicine, and preventive medicine. Each question contains four candidate answers labeled A–D. Compared with MedQA, MedMCQA covers a wider range of medical specialties and contains a larger proportion of knowledge-oriented examination questions.

DDXPlus. DDXPlus is a large-scale synthetic dataset designed for automatic medical diagnosis (Fansi Tchango et al., 2022). Each case describes a patient through demographic information, symptoms, medical history, and other relevant clinical evidence. In the multiple-choice formulation used in our experiments, the model must identify the ground-truth pathology from six candidate diagnoses labeled A–F. Unlike the medical examination datasets, DDXPlus primarily evaluates case-based differential-diagnosis reasoning. All patient cases are synthetically generated and contain no real patient records.

A.2 Math Datasets

AMC. The AMC data used in our controlled answer-space experiment are obtained from the publicly released code repository accompanying TTRL (Zuo et al., 2026). The dataset contains 83 problems from the American Mathematics Competitions, originally presented as open-ended questions with numerical answers.

AMC-MCQ construction. To construct the matched four-option version, we first use the unadapted Llama-3.1-8B-Instruct policy to generate 32 responses for each open-ended problem at temperature 1.0. We extract the final numerical answer from every response and identify the three most frequent incorrect answers produced by the policy. These three model-generated errors are then combined with the ground-truth answer to form four candidate options. The option positions are independently randomized for each problem using seed 1000.

We use model-generated errors rather than randomly sampled distractors because arbitrary distractors may lie in regions of the answer space that the policy rarely visits. In contrast, the selected distractors represent errors that the policy naturally produces, allowing independently generated incorrect reasoning trajectories to collide on the same wrong option. The original and multiple-choice variants therefore contain the same mathematical problems and use the same policy and training configuration, while differing in their response format and answer-space structure.

B Evaluation Metrics

Notation. For each question $q \in \mathcal{D}$, the policy independently generates $K = 16$ rollouts. Following the notation in the main paper, rollout k contains a sequence of reasoning steps followed by a

final answer $a_{q,k}$. We use $a_q^\star$ to denote the ground-truth answer and $\hat{a}_q$ to denote the majority-vote answer. A response whose final answer cannot be parsed is treated as incorrect.

Average accuracy over 16 samples. We define the average accuracy over K sampled rollouts as

$$\text{avg@}K = \frac{1}{|\mathcal{D}|K} \sum_{q \in \mathcal{D}} \sum_{k=1}^{K} \mathbf{1}\left[a_{q,k} = a_q^\star\right], \quad K = 16 \tag{7}$$

For each question, avg@16 computes the fraction of correct answers among the 16 sampled rollouts and then averages this fraction over the evaluation set. We use avg@16 as the primary sample-level metric because it directly measures the probability that a response sampled from the policy is correct. Unlike voting or reward-model selection, it cannot improve merely by selecting a better response from a fixed sample set.

Majority-vote accuracy. For each question, the majority-vote answer is

$$\hat{a}_q = \text{MV}\left(a_{q,1}, \ldots, a_{q,K}\right). \tag{8}$$

The corresponding majority-vote accuracy is

$$\text{maj@}K = \frac{1}{|\mathcal{D}|} \sum_{q \in \mathcal{D}} \mathbf{1}\left[\hat{a}_q = a_q^\star\right], \quad K = 16 \tag{9}$$

Thus, maj@16 performs one majority vote over all 16 responses for each question. This metric measures the accuracy of the single consensus answer produced under a 16-response sampling budget. We use it when comparing PROSE with inference-time selection methods, which also process 16 responses and return one final answer.

Pseudo-label accuracy. In standard TTRL, $\hat{a}_q$ is used as the pseudo-label. For a training batch $\mathcal{B}$, its accuracy is

$$\text{LabelAcc} = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \mathbf{1}\left[\hat{a}_q = a_q^\star\right] \tag{10}$$

Label accuracy measures whether the question-level pseudo-label used to construct the TTRL reward is correct. The ground-truth answer is used only to monitor this metric and is never used to construct the training reward.

Reward accuracy. For rollout k of question q, the majority-vote reward and the corresponding oracle reward are

$$\begin{aligned} r_{q,k}^{\text{MV}} &= \mathbf{1}\left[a_{q,k} = \hat{a}_q\right] \\ r_{q,k}^{\text{GT}} &= \mathbf{1}\left[a_{q,k} = a_q^\star\right] \end{aligned} \tag{11}$$

Reward accuracy measures the proportion of rollouts for which these two binary reward decisions agree:

$$\text{RewardAcc} = \frac{1}{|\mathcal{B}|K} \sum_{q \in \mathcal{B}} \sum_{k=1}^{K} \mathbf{1}\left[r_{q,k}^{\text{MV}} = r_{q,k}^{\text{GT}}\right] \tag{12}$$

Reward accuracy evaluates the correctness of the rollout-level training signal rather than only the correctness of the question-level pseudo-label. Even when $\hat{a}_q$ is wrong, the two rewards agree on an incorrect rollout that matches neither $\hat{a}_q$ nor $a_q^\star$, because both rewards assign it zero.

Protection gap. The protection gap is defined as

$$\text{ProtectionGap} = \text{RewardAcc} - \text{LabelAcc} \tag{13}$$

A positive protection gap means that rollout-level reward assignments are more reliable than the pseudo-labels from which they are derived. In a large open-ended answer space, incorrect rollouts are dispersed across many distinct answers. Even when the pseudo-label is wrong, most other incorrect rollouts do not match it and therefore avoid receiving a false positive reward. In a small multiple-choice answer space, incorrect rollouts frequently collide on the same wrong option, causing more flawed trajectories to receive positive reward and reducing the protection gap.

Algorithm 1 PROSE, one optimization step

Input: question q, policy π_θ, process model MED-PRM

Sample K rollouts $\{o_k\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid q)$
for each rollout o_k **do**
  Score steps $p_i^{(k)} \leftarrow \text{MED-PRM}(s_i^{(k)})$
  Set $r_k \leftarrow \min_i p_i^{(k)}$
  if a_k is not parseable **then** $r_k \leftarrow 0$
end for
Update θ with GRPO using rewards $\{r_k\}_{k=1}^{K}$

C Implementation and Training Details

C.1 PROSE Algorithm

Algorithm 1 summarizes one PROSE optimization step as described in Section 4.

C.2 Models

The main experiments use Llama-3.1-8B-Instruct as the policy model (Grattafiori et al., 2024). The model-scale study additionally evaluates Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Yang et al., 2025). All policies are initialized from their publicly released instruction-tuned checkpoints. We perform full-parameter fine-tuning rather than parameter-efficient adaptation (LoRA rank zero). Policy parameters and vLLM rollouts use BF16 precision, while gradient reduction and model buffers use FP32.

The process reward model is Med-PRM (Yun et al., 2025). It is loaded as a complete checkpoint with its language-modeling head, kept frozen throughout adaptation, and used only to score the sampled reasoning trajectories. No gradient is propagated through Med-PRM. Retrieval follows the configuration released with Med-PRM (Yun et al., 2025): we use the same document corpus, retriever, and number of retrieved documents, and do not modify the retrieval pipeline. During each training run, seven GPUs are allocated to policy training and rollout generation, while one dedicated GPU hosts Med-PRM.

For all Qwen3 policies, thinking mode is disabled. This produces the same explicit step-by-step response format used by Llama-3.1-8B-Instruct and ensures that the reasoning-step markers required by Med-PRM are consistently defined.

C.3 Test-set separation

The evaluation questions used in this work are drawn from the same held-out test datasets used to evaluate Med-PRM and are not included in the data used to train Med-PRM. Likewise, the MedQA-5op and MedQA-4op subsets used here contain no overlapping questions, so the cross-dataset transfer results do not reuse adaptation items. Moreover, neither the test answers nor their ground-truth labels are provided to Med-PRM during scoring. Instead, Med-PRM conditions its step-level judgments on medical evidence retrieved from external clinical guidelines and literature. It uses this evidence to assess whether each reasoning step is logically and clinically supported. In PROSE, the resulting scores provide rewards for policy adaptation; in the inference-time BoN and SC+RM baselines, they are used to identify better-supported responses.

C.4 Prompt Formats

Two distinct prompts are used in our pipeline. The policy prompt elicits a step-by-step solution and a final answer, whereas the Med-PRM prompt asks the reward model to evaluate individual reasoning steps. Each model is given the prompt format used by its own preprocessing and chat template.

Clinical policy prompt. The same policy prompt is used during training and evaluation for all clinical datasets. The system instruction is shown below.

> Solve the following medical question step-by-step. Each step of your explanation must start with the 'Step {number}:' format. You must provide the answer using the phrase 'the answer is (option alphabet)' at the end of your response.

The string `{number}` appears literally in the instruction and is not filled as a template variable. The user message consists of the question stem, a blank line, and one option per line:

```
{question stem}
(A) {option 1}
(B) {option 2}
(C) {option 3}
(D) {option 4}
...
```

If a question stem already contains a lettered option block, the options are not appended again. Option letters are assigned according to position, and any letter prefix already present in the source option string is removed before formatting. This prevents duplicated prefixes such as "(A) (A) aspirin."

AMC policy prompts. The open-ended AMC condition uses the prompt released with the AMC-TTT split in the public TTRL repository. It has no system message; instead, the following instruction is appended to the user turn:

```
{problem statement}
Please reason step by step, and put your final answer within \boxed{}.
```

The output is a free-form numerical answer and is graded using TTRL's math grader after numerical normalization. For the four-option condition, the original boxed-answer suffix is removed and the policy receives the following system instruction:

> Solve the following math question step-by-step. You must provide the answer using the phrase 'the answer is (option alphabet)' at the end of your response.

Its user message contains the same mathematical problem followed by four options in the same one-option-per-line format used for the clinical datasets. Option order is independently shuffled for each problem using seed 1000, and the resulting position of the correct numerical answer is stored as the answer label.

Med-PRM prompt. Med-PRM receives a separate evaluator prompt rather than the answer-generation prompt used by the policy. All reported experiments use the retrieval-augmented system instruction below.

> You are an evaluator assessing the logicality and validity of the reasoning in each step of the given explanation. In order to support the evaluation, the relevant documents, the question, and the explanation are provided sequentially. If the reasoning contains errors, output - after that step. If the reasoning in a step is logical and valid, output + after that step.

The corresponding user message contains the retrieved documents, question, and step-marked explanation in that order:

```
Document 1: {retrieved passage 1}
Document 2: {retrieved passage 2}
...
Question: {question stem} (A) {option 1} (B) {option 2} ...
Explanation: Step 1: {reasoning} [STEP-MARKER] Step 2: {reasoning}
[STEP-MARKER] ...
```

Setting	Value
Algorithm	GRPO; $\gamma = \lambda = 1$
Batch per step	7 prompts $\times$ 32 rollouts = 224 responses
Learning rate	5×10^{-7}
Optimizer	AdamW; $\beta = (0.9, 0.999)$; weight decay 0.01
LR schedule	Cosine; 3% warm-up; minimum-LR ratio 0
PPO clipping	Symmetric clipping with $\epsilon = 0.2$
PPO epochs	1 per optimization step
KL regularization	`low_var_kl`; coefficient 10^{-3}
Gradient clipping	1.0
Training rollouts	Temperature 1.0; top-p 1.0; top-k disabled
Maximum prompt length	1,024 tokens
Maximum response length	2,048 tokens
Maximum vLLM context	3,072 tokens
Sharding	FSDP with parameter and optimizer CPU offload
Gradient checkpointing	Enabled
Attention	FlashAttention-2; padding removed
vLLM	Tensor parallelism 1; GPU memory utilization 0.43–0.50

Table 5: Training configuration used for test-time adaptation.

Unlike the policy prompt, the options in the Med-PRM prompt are placed inline after the question stem. Newline characters in the policy response are replaced with spaces before step marking, so segmentation relies on the `Step <number>:` pattern rather than on line boundaries. In the box above, `[STEP-MARKER]` denotes the model-specific marker token appended by the implementation to every detected step.

Both the policy and Med-PRM messages are wrapped with the chat template of the corresponding model, with the generation prompt appended. The Med-PRM input is capped at 4,096 tokens. A budget of 1,024 tokens is reserved for the question and explanation, and retrieved documents are truncated from the end as needed. The policy input is capped at 1,024 tokens and the generated response at 2,048 tokens, giving a total policy context length of 3,072 tokens.

C.5 Training Configuration

We optimize the policy using critic-free GRPO (Shao et al., 2024). Advantages are normalized by the standard deviation within each rollout group. Table 5 lists the resolved training configuration used in the reported experiments.

The KL term is applied as a loss penalty rather than as part of the rollout reward: `use_kl_in_reward=False`, and the reward-side KL coefficient is zero. PROSE uses the minimum Med-PRM step probability as the trajectory reward. Consistent with the format guard, this reward is overridden to zero when no final answer can be parsed.

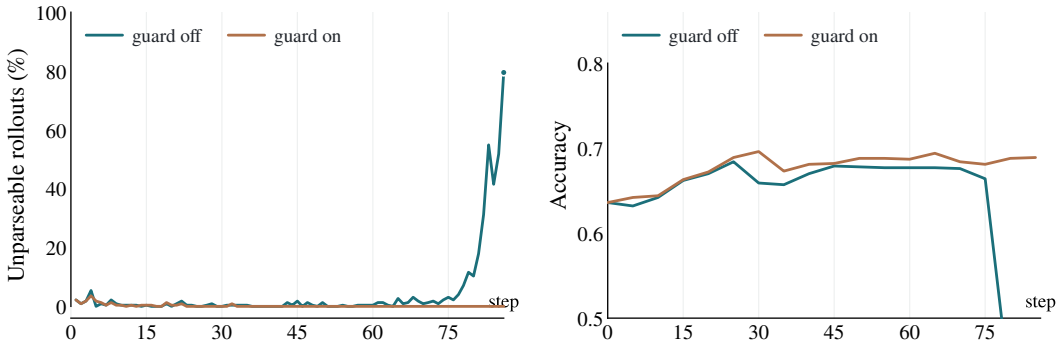
C.6 Evaluation Configuration

For every question, we sample 16 responses using temperature 0.6 and top-p 0.95. This follows the sampling configuration used by TTRL (Zuo et al., 2026). Evaluation is performed every five optimization steps. Step 0 corresponds to the unadapted base policy before any parameter update.

TTRL, PROSE, BoN, and SC+RM are evaluated with the same generation budget of $K = 16$ responses per question. BoN and SC+RM reuse these 16 responses and do not perform additional policy generation. BoN selects the response with the highest Med-PRM score, whereas SC+RM selects the answer whose supporting responses have the largest summed Med-PRM score. Unlike PROSE, both selection baselines require Med-PRM scoring at inference time.

C.7 Random Seeds

For Figure 1, both TTRL and PROSE are independently trained using the same three rollout seeds: 0, 1000, and 2000. Across these runs, PROSE achieves an avg@16 of 0.720 ± 0.004 and a maj@16 of



(a) Format-failure rate (b) Avg@16

Figure 6: **Effect of the format guard. (a)** Without the guard, the fraction of unparseable rollouts rises sharply late in training, whereas it remains close to zero with the guard. **(b)** The increase in format failures causes avg@16 to collapse; enabling the guard maintains stable accuracy.

0.732 ± 0.007, whereas TTRL achieves an avg@16 of 0.675 ± 0.014 and a maj@16 of 0.686 ± 0.009. These values are reported as the mean $\pm$ standard deviation across the three seeds.

All other experiments, including the main result tables, the model-scale study, the controlled answer-space experiment, the reward-aggregation ablation, and the out-of-distribution evaluation, use rollout seed 1000 unless otherwise stated.

D Additional Results

Effect of the Format Guard The process reward model evaluates the marked reasoning steps but does not directly score whether the response ends with a parseable final answer. Consequently, without an explicit constraint, the policy may continue receiving a high process reward while gradually abandoning the required answer format. To test whether the format guard prevents this failure mode, we conduct a controlled ablation in which the two runs differ only in whether an unparseable response has its trajectory reward overridden to zero.

As shown in Figure 6, the two configurations behave similarly during the early stages of optimization. Without the format guard, however, the fraction of unparseable rollouts begins to increase rapidly after approximately 75 optimization steps and eventually reaches about 80%. This behavior is accompanied by a sharp collapse in avg@16. In contrast, enabling the guard keeps the format-failure rate close to zero.

These results demonstrate that the format guard is not merely a post-processing heuristic. It closes a blind spot in the process reward: a rollout cannot retain a positive reward by producing plausible reasoning without a valid terminal answer. Although format failures are rare during most of training, the guard prevents them from becoming a reward-exploiting strategy under prolonged optimization.