When Reward-Free Signals Are Not Directions: A Gradient-Level Diagnostic for Test-Time Reasoning Optimization

Shijie Zheng[1,2], **Yinxing Dai**[2,3], **Zhiqiang Pu**[2,3]
[1]School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences
No. 1 Yanqihu East Road, Huairou District, Beijing 101408, China
[2]National Key Laboratory of Cognition and Decision Intelligence for Complex Systems
Institute of Automation, Chinese Academy of Sciences
No. 95 Zhongguancun East Road, Haidian District, Beijing 100190, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences
No. 1 Yanqihu East Road, Huairou District, Beijing 101408, China
zhengshijie2025@ia.ac.cn daiyinxing2026@ia.ac.cn
Correspondence: zhiqiang.pu@ia.ac.cn

Abstract

First-order test-time optimization (TTO) updates continuous variables such as logits or latent states for a single input. Successful methods usually rely on external reward models or verifiers to supply a semantic gradient. We ask whether reward-free signals from model distributions or prompted self-verification can supply comparable update directions for reasoning tasks. In logit-space TTO, we compare each reward-free gradient with an operational reward-model reference direction, instantiated as the reward-model gradient on the same logits. Across Qwen2.5-7B-Instruct, Qwen3-14B, and Llama-3.1-8B-Instruct, confidence gradients remain nearly orthogonal to the reference, entropy-slope gradients have vanishing scale near greedy rollouts, and terminal answer certainty is sparse and reference-sensitive. Prompted-verifier gradients are more reference-dependent: in a tokenizer-matched Llama diagnostic, four variants attain moderate RM-relative alignment and measurable local RM-score gains. Nevertheless, in end-to-end TTO on Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, the selected reward-free signals produce no wrong-to-correct flips on MATH-500, AIME-2024, AIME-2025, or GSM8K, while a separate reward-model-guided TTO reproduction provides an external positive control with measurable gains.

1 Introduction

Language models can improve reasoning by spending additional computation at inference time. Common approaches include sampling multiple solutions (Wang et al., 2023), searching over intermediate reasoning states (Yao et al., 2023), reranking candidates with a verifier (Lightman et al., 2023; Snell et al., 2025), or revising an answer using self-feedback (Madaan et al., 2023). These methods use more computation for the current input instead of changing model parameters.

First-order test-time optimization (TTO) is a related but more specific approach. It updates a continuous representation of the current response using gradients. If the gradient direction is aligned with reasoning quality, the method can refine a candidate solution without enumerating many complete alternatives. Recent reward-guided continuous-space methods show that inference-time optimization can be useful when guided by external reward models, verifiers, or reward-based control objectives (Yuan et al., 2026; Wang et al., 2026; Kanai et al., 2026).

The practical limitation is the reward signal. External reward models add cost, require training data, and may not be available for a new domain. Here, *reward-free* denotes test-time objectives derived from the model's own distributions or prompted self-verification, without an external reward model or task verifier during optimization. Such signals are a natural alternative: correct solutions often have higher confidence, lower uncertainty, clearer final-answer distributions, or higher self-verification scores (Chen et al., 2023; Wang et al., 2024; Taubenfeld et al., 2025; Kang et al., 2025; Zhu et al., 2026). They motivate the question we study:

> Can reward-free distributional and prompted-verifier signals provide reliable first-order directions for test-time reasoning optimization?

The answer is not implied by cross-sample correlations. A score can distinguish correct and incor-

Code: reward-free-tto-diagnostics

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 24687–24703
October 24-29, 2026 ©2026 Association for Computational Linguistics

rect answers across a dataset without giving a useful local direction for a particular answer. A verifier may assign higher scores to correct complete solutions, yet its local gradient around a flawed solution may make the text look more verifier-friendly rather than more correct. Similarly, increasing confidence in a current solution may reinforce the existing trajectory instead of repairing its error. We study this gap between a score and a direction.

We use logit-space TTO as a controlled setting because generated-token logits provide shared differentiable coordinates for a candidate signal and the reward-model (RM) reference. For an initial solution with logits z, we compare the gradient of an internal signal $f(z)$ with the gradient of an external RM with respect to the same logits. We test whether a candidate reproduces an empirically useful RM-guided local direction, and separately evaluate whether optimizing it improves task correctness. The empirical findings concern this logit-space setting; the diagnostic can be instantiated in another differentiable space when both objectives act on the same variables.

We make three contributions:

1. We formulate a gradient-level evaluation protocol for first-order TTO signals, combining cosine alignment, norm ratio, and end-to-end TTO validation.

2. We compare distributional confidence, entropy slope, answer certainty, and prompted self-verification signals under the same logit-space diagnostic.

3. We identify a signal-specific separation in mathematical reasoning: distributional objectives fail through near-orthogonality, vanishing scale, or sparse and unstable support, while prompted verifiers can provide moderate local RM-relative directions without yielding end-to-end corrections under the tested greedy logit-space optimizer.

2 Related Work

Test-time computation for language models includes candidate sampling, reranking, search, refinement, verifier-guided decoding, and continuous-state updates (Snell et al., 2025; Wang et al., 2023; Li et al., 2025; Yuan et al., 2026; Wang et al., 2026). We use test-time optimization in a narrower sense: additional computation at inference time optimizes a response, prompt, logits, latent state, or reasoning path for the current input. This excludes general test-time adaptation unless it directly optimizes the current response.

We organize related work by two axes: whether the signal is supervised or reward-free, and whether the method uses zero-order candidate evaluation or first-order gradient updates. This yields four cases. Reward-free zero-order methods use internal signals to sample, rank, select, or revise candidates without differentiating the signal. Tree-of-Thoughts uses LM self-evaluation during search (Yao et al., 2023); Self-Refine revises outputs with self-generated feedback (Madaan et al., 2023); and Universal Self-Consistency selects among candidates using the model itself (Chen et al., 2023). Other work uses likelihood, confidence, self-certainty, or entropy dynamics for selection (Wang et al., 2024; Taubenfeld et al., 2025; Kang et al., 2025; Zhu et al., 2026). These methods show that internal signals can score or filter complete answers, but they do not test whether the gradient of the same signal improves a single answer.

Supervised or verifier-guided zero-order methods use external semantic feedback to score candidates. Outcome reward models, process reward models, generative verifiers, and process advantage verifiers have been used for best-of-N selection, beam search, and tree search (Cobbe et al., 2021; Lightman et al., 2023; Snell et al., 2025; Setlur et al., 2024; Zhang et al., 2025a; Khalifa et al., 2025). This line of work motivates our operational RM reference: if external reward models can guide search, their gradients are a reasonable operational reference for first-order methods. Related self-correction studies also find that intrinsic feedback alone often fails without external feedback (Huang et al., 2024; Zhang et al., 2024).

Supervised first-order TTO directly updates continuous variables using reward-model or verifier gradients. SEA optimizes responses in continuous latent or logit space (Yuan et al., 2026); ∇-Reasoner combines decoding with reward-model gradients (Wang et al., 2026); and AISP gives a related pre-logit-space approach based on sampling-based optimal control (Kanai et al., 2026). These methods show that first-order TTO can work when the direction comes from external semantic supervision. We do not argue against first-order TTO; we ask whether that external gradient can be replaced by reward-free internal or self-verification gradients.

The remaining case is reward-free first-order optimization. Prior work has explored sample-specific hidden-state adaptation, latent-space self-reward optimization, entropy minimization, and online RL from unlabeled test data (Hu et al., 2025; Li et al., 2026; Agarwal et al., 2025; Zuo et al., 2025; Zhao et al., 2026; Prabhudesai et al., 2025). Recent analysis also suggests that internal-feedback objectives can be fragile across training stages and model types (Zhang et al., 2025b). Unlike reward-free or intrinsically motivated RL, which studies exploration or policy/value learning over repeated environment interactions, our setting locally optimizes one generated response at inference time. We ask whether reward-free distributional and self-verifier signals provide useful local first-order directions. Such a direction can fail even when the same signal is useful for zero-order selection, reranking, self-consistency, or uncertainty estimation.

3 Gradient-Level Diagnostic

Let a model generate an initial solution y for prompt x. We denote the sequence logits by

$$z \in \mathbb{R}^{L\times|V|}, \tag{1}$$

where L is the generated length and $|V|$ is the vocabulary size. A logit-space first-order TTO method updates z by

$$z^{(k+1)} = z^{(k)} + \eta g(z^{(k)}), \tag{2}$$

where k indexes optimization steps.

Differentiable logit path. Evaluator scores are defined on token sequences, but discrete argmax decoding blocks gradients. We therefore represent each generated position as a soft mixture of evaluator-side token embeddings while keeping the prompt and evaluation suffix fixed:

$$e_i(z_i) = \sum_{v\in V_{\text{gen}}} \text{softmax}(z_i/\tau)_v \widetilde{E}^{\text{eval}}_v, \tag{3}$$

where τ is the softmax temperature and $\widetilde{E}^{\text{eval}} \in \mathbb{R}^{|V_{\text{gen}}|\times d_{\text{eval}}}$ is the evaluator embedding matrix aligned to the generator vocabulary.

For self-verifier signals, the evaluator is the generator itself, so the tokenizer and embedding matrix are shared. For reward-model gradients used as RM reference directions, when tokenizers differ, we use a fixed evaluator-side embedding alignment and allow gradients to flow only through generator softmax weights. This cross-tokenizer path is used as an operational reward-gradient construction; Appendix B.2 describes the exact token-string matching procedure and reports match coverage.

Prompt tokens and evaluation-suffix tokens are kept fixed, and gradients are taken only with respect to generated-token logits. The evaluator receives the fixed prompt embeddings, the soft generated embeddings, and the fixed evaluation suffix. This gives a differentiable path from reward or verifier score back to z without treating the discrete decoded string as differentiable.

For reward-model-based TTO, the direction is

$$g_{\text{RM}} = \nabla_z r_{\text{RM}}(z). \tag{4}$$

For an internal signal f, let the raw gradient be

$$\tilde{g}_f = \nabla_z f(z). \tag{5}$$

The candidate update direction is the signed gradient

$$d_f = \sigma_f \tilde{g}_f, \tag{6}$$

where $\sigma_f = +1$ for signals we maximize and $\sigma_f = -1$ for signals we minimize. The question is whether d_f can replace g_{RM}.

First-order interpretation. The diagnostic is local with respect to the reward-model score. Let

$$\Delta r_{\text{RM}} = r_{\text{RM}}(z + \eta d_f) - r_{\text{RM}}(z). \tag{7}$$

For a small step η along d_f, the first-order expansion gives

$$\begin{aligned}\Delta r_{\text{RM}} &= \eta\langle g_{\text{RM}}, d_f\rangle + O(\eta^2\|d_f\|^2)\\ &= \eta\|g_{\text{RM}}\|\|d_f\|\cos(d_f, g_{\text{RM}})\\ &\quad + O(\eta^2\|d_f\|^2).\end{aligned} \tag{8}$$

With an L_r-Lipschitz reward gradient, this gives

$$\begin{aligned}\Delta r_{\text{RM}} &\geq \eta\|g_{\text{RM}}\|\|d_f\|\cos(d_f, g_{\text{RM}})\\ &\quad - \frac{L_r\eta^2}{2}\|d_f\|^2.\end{aligned} \tag{9}$$

Thus cosine controls the first-order reward-score change, while the norm ratio controls whether the signal has enough scale to affect the update. Equation (8) is local for sufficiently small steps, and Equation (9) assumes an L_r-Lipschitz RM gradient to bound the higher-order remainder. A weak or negative cosine gives little first-order reward-score increase and therefore cannot serve as a reliable local replacement for the reward-model direction. A very small norm yields little movement under

the fixed optimization scale, even when the sign is favorable. This diagnostic is relative to the reward-model reference direction; it does not assume access to, or make claims about, the ground-truth correctness gradient.

For end-to-end TTO with signal s, we maximize the following objective over the same generated logits:

$$J_s(z) = \sigma_s \lambda s(z) - \beta \,\mathrm{NLL}(y_0; z), \tag{10}$$

where y_0 is the greedy initial solution, σ_s is positive for signals we maximize and negative for signals we minimize, and the NLL term regularizes the optimized logits toward the original model-supported response. Each run greedily decodes an initial solution, performs K gradient updates on generated-token logits, decodes the optimized logits by argmax, and evaluates the extracted final answer.

We use the reward-model gradient as an operational RM reference direction. This choice follows recent reward-guided first-order TTO methods that optimize continuous variables using reward-model or verifier feedback (Yuan et al., 2026; Wang et al., 2026). The RM reference is operational rather than ground-truth: it is not a true mathematical correctness gradient, but a semantic reference direction supported by reward-guided TTO and reward-score sanity checks. When the positive control comes from an external implementation, we treat it as evidence that reward-guided first-order optimization can work, not as certification of every implementation detail in our logit-space pipeline.

We evaluate a candidate signal with two quantities. The first is cosine alignment with the RM reference direction:

$$\cos(d_f, g_{\mathrm{RM}}) = \frac{\langle d_f, g_{\mathrm{RM}} \rangle}{\|d_f\| \|g_{\mathrm{RM}}\|}. \tag{11}$$

The second is the norm ratio:

$$\rho(f) = \frac{\|d_f\|}{\|g_{\mathrm{RM}}\|}. \tag{12}$$

Cosine measures whether the signal points in a similar direction to the reward-model gradient. The norm ratio measures whether the signal has usable magnitude. A near-zero cosine indicates that the signal may describe useful properties without reproducing the RM-guided local direction. A very small norm ratio indicates that the signal is unlikely to drive the update. A large norm with near-zero cosine can move logits by a large amount without moving toward the reward-model direction.

All alignment scores are computed per example by flattening the signed generated-token logit direction over $L \times |V|$ and then averaging across examples. Prompt-token gradients are excluded. For answer-certainty signals, whose support is concentrated near the answer position, we additionally check whether the sparse gradient remains stable under end-to-end TTO rather than relying on full-sequence cosine alone.

The labels in the result tables are reporting shorthand rather than decision rules. Our conclusions rely on the continuous cosine and norm-ratio values, the random-direction baseline, the directional-derivative sanity check, and the end-to-end wrong-to-correct flip counts. When using labels, we call near-zero or degenerate directions infeasible and positive but small alignments limited.

This sign convention follows the actual optimization direction: for a minimized signal, $d_f = -\nabla_z f(z)$; for a maximized signal, $d_f = +\nabla_z f(z)$.

We instantiate this diagnostic on four reward-free signal families: local confidence, entropy-trajectory slope, terminal answer-token certainty, and prompted self-verification. The first three are distributional signals, while the last uses the full generated solution through a verifier prompt.

Confidence. Confidence-based scores have been used for self-consistency weighting, best-of-N selection, and internal-feedback learning (Taubenfeld et al., 2025; Kang et al., 2025; Zhao et al., 2026; Prabhudesai et al., 2025). The confidence objective is the average negative log probability over top-m tokens:

$$C(z_i) = -\frac{1}{m} \sum_{v \in \text{top-}m(P_i)} \log P_i(v), \tag{13}$$

$$P_i = \mathrm{softmax}(z_i). \tag{14}$$

With a fixed top-m set, its gradient at position i has the form

$$\frac{\partial C}{\partial z_{ij}} = P_i(j) - \frac{1}{m} \mathbf{1}[j \in \text{top-}m]. \tag{15}$$

Lower C assigns more probability mass to the current top-m alternatives, so TTO uses $-\nabla C$ as the confidence-improving direction. For small m, this encourages local concentration around the current high-probability alternatives. This gradient reshapes local token probabilities, but it does not encode whether the reasoning trajectory is correct.

Signal	Meaning	Direction	Scope	Full TTO
C	top-m confidence loss	minimize	all generated positions	yes
S	entropy-decrease slope	maximize	all generated positions	yes
SS	terminal answer uncertainty	minimize	answer position	yes
V_A	answer-correctness log-odds	maximize	full response	yes
V_B	error-free log-odds	maximize	full response	diagnostic only
V_A^{ND}	format-controlled correctness	maximize	full response	diagnostic only
V_A^{INJ}	injected-answer control	maximize	full response	diagnostic only
V_C	process-validity log-odds	maximize	full response	yes

Table 1: Reward-free signals evaluated in the gradient diagnostic. Full TTO uses the selected signals marked "yes."

Entropy slope. Entropy and entropy dynamics have recently been used as uncertainty signals for reasoning selection and as unsupervised learning or inference-time objectives (Zhu et al., 2026; Agarwal et al., 2025; Prabhudesai et al., 2025). Entropy slope uses the trajectory of token entropy. Let H_i be the entropy at position i. We define the slope by a least-squares trend over generated positions,

$$S(z) = -\frac{\sum_i (i - \bar{i})(H_i - \bar{H})}{\sum_i (i - \bar{i})^2}, \qquad (16)$$

so larger S corresponds to a steeper entropy decrease. The entropy gradient contains

$$\frac{\partial H_i}{\partial z_{ij}} = -P_i(j)(H_i + \log P_i(j)). \qquad (17)$$

Near greedy rollouts, token distributions are often peaked. For the selected token j^*, $P_i(j^*)$ is close to 1, H_i is close to 0, and $\log P_i(j^*)$ is close to 0. This makes the entropy gradient small.

Answer certainty. Final-answer certainty is related to recent self-certainty and confidence-based selection objectives (Kang et al., 2025; Taubenfeld et al., 2025; Zhao et al., 2026). We implement this signal as an answer-uncertainty loss over the final answer distribution:

$$SS(z) = 1 - \max_v P_{\mathrm{answer}}(v). \qquad (18)$$

Lower SS means higher terminal-token certainty; gradients are taken through the current maximum-probability answer token. In the math benchmarks, the answer position is obtained from the extracted final answer span when available; otherwise the final generated token is used as a fallback. Multi-token answers are therefore approximated by their terminal answer token in this diagnostic. This is a deliberately simple terminal-token diagnostic rather than a full answer-string confidence estimator. The signal is sparse: it acts near the answer position and provides little direct signal for earlier reasoning steps. It also saturates when the model is already confident. For a confidently wrong answer, increasing certainty may reinforce the wrong final state.

Self-verifier signals. Self-verification and self-correction have been widely studied as test-time feedback mechanisms, but prior work finds that intrinsic self-correction without external feedback is often unreliable (Madaan et al., 2023; Huang et al., 2024; Zhang et al., 2024). Self-verifier signals ask the generator itself to judge the generated solution. A typical signal is a log-odds score:

$$V_A(z) = \log P(\mathrm{correct} \mid x, y(z)) - \log P(\mathrm{incorrect} \mid x, y(z)). \qquad (19)$$

We evaluate five verifier variants: answer correctness (V_A), error detection (V_B), a format control that avoids direct-answer cues (V_A^{ND}), an injected-answer control (V_A^{INJ}), and process validity (V_C). These signals use the full generated solution through the same soft-embedding path described in Section 3, and therefore contain more semantic information than local distributional signals. They are also sensitive to prompt wording, answer format, and self-evaluation calibration.

4 Experimental Setup

Models. We evaluate Qwen2.5-7B-Instruct (Yang et al., 2025b) as the main model for gradient alignment and full TTO, Qwen3-14B (Yang et al., 2025a) as a larger Qwen-family gradient-alignment validation model, and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as a non-Qwen validation model. We reserve full end-to-end TTO evaluation for Qwen2.5-7B and Llama-3.1-8B, where the selected end-to-end TTO signals were run under the same evaluation protocol. To isolate tokenizer mismatch in the gradient

diagnostic, we additionally pair Llama-3.1-8B-Instruct with Skywork-Reward-V2-Llama-3.1-8B on the fixed 50-example MATH-500 subset; this matched path uses token-ID identity mapping.

Datasets. The main math diagnostic set contains 110 problems: 50 from MATH-500 (Hendrycks et al., 2021), 30 from AIME-2024, and 30 from AIME-2025. GSM8K (Cobbe et al., 2021) gradient alignment uses a 150-example subset from the official test split: we rank examples by a deterministic heuristic based on question length, solution length, number count, arithmetic-operator count, and sentence count, split the ranked set into easy, medium, and hard thirds, and sample 50 examples from each bucket with seed 42. These fixed subsets are used for the compute-intensive gradient diagnostic. End-to-end TTO is evaluated on 560 MATH/AIME examples and the full 1,319-example GSM8K test set for each of Qwen2.5 and Llama. Across the four model–dataset settings, each full-TTO signal is evaluated on 3,758 outputs, including 1,043 initially incorrect outputs eligible for correction.

TTO hyperparameters. The standard TTO setting uses $K = 20$ gradient steps, learning rate 0.01, $\lambda = 0.1$, and negative-log-likelihood coefficient 0.001. We also test an aggressive V_C setting with $K = 100$, learning rate 0.01, $\lambda = 1.0$, and the same negative-log-likelihood coefficient. Initial responses are generated by greedy decoding. Correctness is evaluated with a fixed answer-extraction and grading pipeline. The pipeline extracts the final model answer and applies mathematical equivalence grading for MATH/AIME; for GSM8K, gold answers are normalized from the final #### field by removing commas and taking the last numeric value before numeric matching. The final TTO output is obtained by argmax decoding from the optimized logits.

Implementation details. Unless otherwise stated, confidence uses top-$m = 10$, the soft-embedding path uses temperature $\tau = 1$, and greedy generation is capped at 512 new tokens. Gradient diagnostics use at most the first 512 generated tokens and exclude prompt-token gradients. Random-direction baselines sample isotropic directions in the generated-token logit space after the same prompt exclusion, using 100 random directions per example; for the terminal answer-certainty diagnostic, random directions are sampled in the corresponding answer-position logit space. Directional-derivative checks L2-normalize both g_{RM} and random directions to unit norm before applying perturbations.

5 Results

5.1 Gradient Alignment

Table 2 shows the main MATH-500 gradient-alignment results for Qwen2.5-7B-Instruct under the original Qwen-family RM reference. The distributional signals do not satisfy the diagnostic. C has a large norm ratio but near-zero cosine. S has a small positive cosine but a very small norm ratio. SS is sparse and unstable. Some verifier signals have higher alignment; the best value in this reference setup is V_C at 0.242.

Appendix C.1 reports example-level standard deviations, medians, interquartile ranges, and bootstrap confidence intervals for the MATH-500 diagnostic subset. These uncertainty estimates do not change the qualitative interpretation.

The first verifier experiment found that V_A is affected by answer format. On MATH-500, the correct-versus-incorrect log-odds gap is about 8.57, while the boxed-versus-no-boxed gap is about 8.76. Under the original Qwen-family RM reference, the control variants remain modest: V_A^{ND} reaches 0.216 cosine, V_A^{INJ} drops to 0.024, and V_C reaches 0.242.

Table 3 compares Qwen2.5, Qwen3, and Llama on the MATH-500 diagnostic subset under the original Qwen-family RM reference. Confidence and entropy slope show consistent direction or scale failures, answer certainty is sparse and unstable, and prompted-verifier alignment is at most 0.242 in this reference setup.

A tokenizer-matched Llama diagnostic removes the cross-tokenizer approximation. Confidence and entropy slope remain nearly orthogonal, with mean cosine values of 0.010 and 0.011, while answer certainty remains high-variance and reference-sensitive. In contrast, four prompted-verifier variants attain cosine values of 0.304–0.421 and norm ratios of 0.327–0.485. At normalized $\epsilon = 0.1$, positive verifier perturbations increase the matched RM score in 74%–86% of examples, compared with 100% for the RM-reference direction and 36% for a random direction. These local RM-score effects motivate separate end-to-end validation rather than establishing task-level improvement by themselves. Appendix B.3 reports the full results.

As a posthoc random-direction baseline for the

Signal	n_c/n_ρ	cos	ρ	Label
C	50/50	-0.001	16.56	infeasible
S	50/50	0.013	5.20×10^{-5}	infeasible
SS	19/50	-0.427	34.00	infeasible
V_A	50/50	0.153	1.62	limited
V_B	50/50	0.050	3.08	limited
V_A^{ND}	50/50	0.216	1.49	limited
V_A^{INJ}	50/50	0.024	1.73	limited
V_C	50/50	0.242	3.19	limited

Table 2: Qwen2.5-7B-Instruct gradient alignment on the MATH-500 diagnostic subset under the original Qwen-family RM reference. n_c and n_ρ denote valid example counts for cosine and norm ratio. For answer certainty, cosine is computed only on examples with non-degenerate answer-position gradients; ρ is reported over all examples.

Model	C cos/ρ	S cos/ρ	SS cos/ρ	Best verifier cos/ρ
Qwen2.5-7B	-0.001 / 16.56	0.013 / 5.20×10^{-5}	-0.427 / 34.00	V_C: 0.242 / 3.19
Qwen3-14B	0.008 / 18.72	0.021 / 3.89×10^{-5}	-0.288 / 20.28	V_A^{ND}: 0.142 / 0.75
Llama-3.1-8B	0.0036 / 10.06	0.0117 / 1.50×10^{-4}	-0.710 / 10.09	V_A: 0.107 / 1.37

Table 3: Multi-model gradient alignment on MATH-500 under the original Qwen-family RM reference. The Llama row uses the cross-tokenizer path described in Appendix B.2.

original RM-reference setup, we compare each observed signal direction with isotropic norm-matched random directions on the MATH-500 50-example subset. Because random cosines concentrate tightly around zero in high-dimensional logit space, we interpret random percentiles together with absolute cosine rather than as a standalone usefulness criterion. The full random-baseline table is reported in Appendix C.2 (Table 8). Distributional signals remain near this random-direction regime in absolute cosine. Verifier signals are usually above the random-direction median, although their absolute cosine is modest under this reference.

Under the original Qwen-family RM reference, the GSM8K gradient-alignment results show the same pattern. Confidence has near-zero cosine despite large norm, entropy slope has very small norm, and answer certainty remains sparse and unstable. Prompted-verifier cosines remain below 0.15 on Qwen2.5 and Llama, and the best Qwen3 value is 0.033. Appendix C.3 reports the full table.

5.2 End-to-End TTO

Table 4 reports MATH/AIME TTO accuracy, flip statistics, and token-change rates for Qwen2.5-7B-Instruct. Across the full-TTO signals in Table 4, every signal has zero wrong-to-correct flips and one correct-to-wrong flip. Mean token-change rates range from 0.65% to 0.75%. Across all four Qwen2.5/Llama model–dataset settings, each full-TTO signal is tested on 1,043 initially incorrect outputs, and none produces a wrong-to-correct transition.

The low token-change rates in Table 4 raise a possible argmax-barrier concern: small continuous logit updates may not change the decoded tokens. We therefore run an aggressive update setting for V_C. In the standard setting, V_C changes 0.74% of decoded tokens on average and produces zero wrong-to-correct flips. In the aggressive setting, the token-change rate increases to 2.23%, but wrong-to-correct flips remain zero. This result points away from insufficient logit movement as the only explanation. It leaves open decoding-aware objectives or search-coupled first-order updates as follow-up work.

The full boundary-check table is reported in Appendix D.3 (Table 12). On GSM8K, Qwen2.5 remains essentially unchanged, from 86.96 greedy accuracy to 86.88–86.90 under the tested TTO signals, with zero wrong-to-correct flips. Llama shows the same pattern on GSM8K, from 78.32 greedy accuracy to 78.24–78.32 after TTO, again with zero wrong-to-correct flips. The completed GSM8K checks are consistent with the Qwen2.5 MATH/AIME result.

Signal	MATH	AIME24	AIME25	Overall	W→C	C→W	Change
Greedy	70.8	6.7	0.0	63.6	–	–	–
C	70.6	6.7	0.0	63.4	0	1	0.65%
S	70.6	6.7	0.0	63.4	0	1	0.67%
SS	70.6	6.7	0.0	63.4	0	1	0.67%
V_A	70.6	6.7	0.0	63.4	0	1	0.75%
V_C	70.6	6.7	0.0	63.4	0	1	0.74%

Table 4: Qwen2.5-7B-Instruct MATH/AIME end-to-end TTO. Accuracy columns are percentages over MATH-500, AIME-2024, AIME-2025, and the combined 560-example set. W→C, C→W, and Change are flip counts and mean token-change rates over the combined set.

5.3 Positive Control and RM-Reference Sanity

We next validate both the effectiveness of reward-guided first-order optimization and the operational relevance of the RM reference. As an external positive control, we use a separate reward-model-guided TTO (RM-TTO) reproduction based on the Nabla-Reasoner implementation (Wang et al., 2026). In this reproduction, RM-TTO improves MATH-500 from 70.8 to 75.4, AIME-2024 from 6.7 to 17.5 mean accuracy, and AIME-2025 from 0.0 to 17.5 mean accuracy. The MATH-500 number comes from one 500-example reproduced run; the AIME numbers are means over four 30-problem reproduced runs. Appendix D.1 reports the AIME per-run accuracies and run-level standard deviations. We use these reproduced runs only as an external positive control showing that reward-guided first-order optimization can improve the same benchmark family; they are not quoted as the official Nabla-Reasoner paper numbers and are not part of our reward-free-signal TTO pipeline.

We also score reproduced outputs with Skywork-Reward-V2-Qwen3-4B (Liu et al., 2026). The continuous reward score is predictive of final correctness: over 740 samples, RM-score AUROC for correctness is 0.866, the correct mean RM score is 17.01, the incorrect mean is 12.35, and the mean score gap is +4.66. Restricted to outputs that contain boxed answers, AUROC remains 0.867, reducing the concern that the reward model only detects answer format.

Re-scoring the same 740 outputs with Skywork-Reward-V2-Llama-3.1-8B gives correctness AUROC 0.823, boxed-only AUROC 0.814, and Spearman correlation 0.830 with the original RM scores. Both RMs therefore provide useful operational references, while individual gradient cosines remain reference-dependent.

Finally, we run a local directional-derivative sanity check on MATH-500. Moving along $+g_{\mathrm{RM}}$ increases the RM score in 92% of Qwen2.5 examples and 90% of Llama examples, while moving along $-g_{\mathrm{RM}}$ decreases the score in 84% and 96% of examples, respectively. Random directions do not show a stable improvement trend (Appendix D.2, Table 11).

The score-level AUROC and local directional-derivative checks support the use of the RM gradient as an operational reference. RM-relative geometry and task correctness remain distinct: the former measures local RM-score motion, while the latter is evaluated independently by end-to-end answer grading.

6 Analysis and Discussion

The experiments separate two statements: correct answers often have property P, and changing the current answer to increase P improves correctness. The first statement can hold while the second fails. Confidence, entropy, and self-verification can be useful for ranking or filtering candidate answers. First-order TTO requires more: the gradient of the signal must point toward a better answer for the current sample.

The confidence gradient depends on local token probabilities. It can sharpen a distribution, but it has no direct access to the global semantic relation between the prompt, reasoning steps, and final answer. This explains why C can have a large norm ratio while remaining nearly orthogonal to the reward-model gradient.

Entropy slope can be informative as a trajectory feature. However, near greedy outputs, token distributions are often peaked. The entropy gradient is then small, which makes S ineffective as a direct update direction. Increasing update strength does not address the reward-model-gradient reference direction problem.

Answer certainty acts near the final answer. It

cannot directly repair earlier reasoning errors. It also has poor coverage when the answer format is missing or ambiguous, and it can reinforce a confident wrong answer.

Prompted-verifier signals behave differently from purely distributional objectives because they inspect the full generated solution. Under the original Qwen-family reference, their alignment is modest and affected by prompt wording and answer format. In the tokenizer-matched Llama diagnostic, however, four variants attain moderate RM-relative alignment and measurable local RM-score gains. This local effect is prompt- and reference-dependent, and the full-TTO evaluations for V_A and V_C still produce no wrong-to-correct transitions. Prompted verification can therefore pass a local RM-relative test without passing the task-level correction test.

The results do not contradict zero-order methods that use internal signals for selection. A score can rank complete candidates even when its local gradient is not a reliable improvement direction. Search and reranking evaluate whole answers; first-order TTO relies on local movement from the current answer. Reward-free first-order TTO may need a signal closer to semantic error correction, for example process-level verification, contrastive self-verification, tool- or constraint-guided feedback, or a hybrid with zero-order search before the first-order update.

7 Conclusion

We studied whether reward-free distributional and prompted-verifier signals can replace reward-model guidance in logit-space first-order TTO for reasoning. Confidence gradients are largely orthogonal to the RM reference, entropy-slope gradients have vanishing scale, and terminal answer certainty is sparse and reference-sensitive. Prompted-verifier gradients can be more informative: in a tokenizer-matched Llama diagnostic, several variants exhibit moderate RM-relative alignment and local RM-score gains. However, these local effects do not translate into wrong-to-correct transitions under the tested greedy logit-space optimizer.

End-to-end TTO complements the gradient diagnostic: moderate local verifier alignment does not translate into answer correction under the tested optimizer. Across MATH/AIME and GSM8K, the tested signals produce no wrong-to-correct flips, while reward-model-based TTO improves performance in the positive-control setting.

The practical implication is that score discrimination, local gradient geometry, and task-level improvement are distinct properties. Future reward-free TTO methods should therefore validate direction, scale, and end-to-end outcomes, not only score correlation or verifier accuracy.

Limitations

The evidence is limited to a specific experimental setting. It does not prove that all reward-free first-order TTO is impossible. The experiments focus on mathematical reasoning. The optimization target is logit-space TTO; results may differ for hidden-state or representation-space methods. The RM reference direction is an operational reward-model gradient, not a ground-truth correctness gradient. Cross-tokenizer alignment may introduce noise in the original Llama comparison. The tokenizer-matched Llama analysis removes this approximation in one non-Qwen setting, although the generator and RM share the same model family; cross-architecture RM independence remains open. The tested self-verifier prompts are not exhaustive. Qwen3-14B is used for gradient-alignment validation rather than full end-to-end TTO; complete full-TTO evidence is reported for Qwen2.5-7B and Llama-3.1-8B. Additional non-Qwen models, larger models, non-math tasks, and alternative verifier prompts would broaden the evidence.

Ethical Considerations

This work evaluates test-time optimization signals on mathematical reasoning benchmarks. It does not introduce new datasets with personal information or deploy a user-facing system. A potential risk is that unreliable internal confidence or self-verification signals may be overinterpreted as semantic correctness signals in high-stakes settings. Our results caution against such use without direct validation.

LLM Usage Statement

Generative AI tools were used for grammar checking, spelling correction, and language polishing. All research ideas, experiments, analyses, and conclusions are the authors' own.

Acknowledgments

This work was supported by the Key Research Project of Chinese Academy of Sciences under Grant No. KGFZD-145-25-21, the National Natural Science Foundation of China under Grant No. 62322316 and 62503472, the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-123), and the Young Scientists Foundation of CSAA (Guidance Navigation and Control, GNC) under Grant CSAA-YSF2025-GNC-08.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *Preprint*, arXiv:2505.15134.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *Preprint*, arXiv:2311.17311.

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. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. 2025. Slot: Sample-specific language model optimization at test-time. *Preprint*, arXiv:2505.12392.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. *Preprint*, arXiv:2310.01798.

Sekitoshi Kanai, Tsukasa Yoshida, Hiroshi Takahashi, Haru Kuroki, and Kazumune Hashimoto. 2026. Test-time reward-guided alignment of language models by importance sampling on pre-logit space. *Preprint*, arXiv:2510.26219.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *Preprint*, arXiv:2502.18581.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *Preprint*, arXiv:2504.16828.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and Zilong Zheng. 2026. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space. *Preprint*, arXiv:2505.13308.

Yafu Li, Xuyang Hu, Xiaoye Qu, Linjie Li, and Yu Cheng. 2025. Test-time preference optimization: On-the-fly alignment via iterative textual feedback. *Preprint*, arXiv:2501.12895.

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

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2026. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *Preprint*, arXiv:2507.01352.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. Maximizing confidence alone improves reasoning. *Preprint*, arXiv:2505.22660.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *Preprint*, arXiv:2410.08146.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pages 10131–10165.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111, Vienna, Austria. Association for Computational Linguistics.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft self-consistency improves language model agents. *Preprint*, arXiv:2402.13212.

Peihao Wang, Ruisi Cai, Zhen Wang, Hongyuan Mei, Qiang Liu, Pan Li, and Zhangyang Wang. 2026. ∇-Reasoner: LLM Reasoning via Test-Time Gradient Descent in Latent Space. In *The Fourteenth International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qwen : An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Yige Yuan, Teng Xiao, Li Yunfan, Bingbing Xu, Shuchang Tao, Yunqi Qiu, Huawei Shen, and Xueqi Cheng. 2026. Inference-time alignment in continuous space. *Preprint*, arXiv:2505.20081.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025a. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025b. No free lunch: Rethinking internal feedback for llm reasoning. *Preprint*, arXiv:2506.17219.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning. *Preprint*, arXiv:2404.17140.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2026. Learning to reason without external rewards. *Preprint*, arXiv:2505.19590.

Chenghua Zhu, Siyan Wu, Xiangkang Zeng, Zishan Xu, Zhaolu Kang, Yifu Guo, Yuquan Lu, Junduan Huang, and Guojing Zhou. 2026. Edis: Diagnosing llm reasoning via entropy dynamics. *Preprint*, arXiv:2602.01288.

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. 2025. Ttrl: Test-time reinforcement learning. *Preprint*, arXiv:2504.16084.

A Artifact and Reproducibility Details

Artifacts. We use publicly released research artifacts: MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AIME benchmark problems, Qwen (Yang et al., 2025b,a) and Llama (Grattafiori et al., 2024) model checkpoints, Skywork-Reward-V2-Qwen3-4B and Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2026), and the Nabla-Reasoner implementation (Wang et al., 2026). We cite the original creators when available, follow the released licenses and terms of use, and do not redistribute model weights, dataset contents, or third-party code. The released repository contains our implementation, experiment configurations, fixed split manifests, and reproducibility documentation. Benchmark contents, model weights, and third-party code are not redistributed.

Data characteristics. The benchmarks are English mathematical reasoning datasets consisting of problem statements and answers. We do not collect new data. We did not identify personally identifying information or offensive content in the examples used in our experiments. Demographic-group coverage is not applicable because the datasets do not contain human-subject attributes.

Data statistics. Section 4 reports the main dataset statistics: the 110-problem math diagnostic set contains 50 MATH-500 problems, 30 AIME-2024 problems, and 30 AIME-2025 problems; GSM8K gradient alignment uses a 150-example subset from the official test split; and end-to-end TTO uses MATH-500, AIME-2024, AIME-2025, and the full GSM8K test set. Appendix tables report valid example counts for each diagnostic.

Implementation and parameters. Section 4 reports the main model and optimization settings. Confidence uses top-$m = 10$, the soft-embedding temperature is $\tau = 1$, greedy generation is capped at 512 new tokens, and gradient diagnostics use at most the first 512 generated tokens. Random-direction baselines use 100 random directions per example. Directional-derivative checks L2-normalize directions before perturbation. Standard TTO uses $K = 20$ steps, learning rate 0.01, $\lambda = 0.1$, and negative-log-likelihood coefficient 0.001; the aggressive V_C setting uses $K = 100$, learning rate 0.01, $\lambda = 1.0$, and the same negative-log-likelihood coefficient.

Evaluation and statistics. Correctness is evaluated with a fixed answer-extraction and grading pipeline. MATH/AIME use mathematical equivalence grading. GSM8K gold answers are normalized from the final #### field by removing commas and taking the last numeric value before numeric matching. Appendix B.1 gives the verifier prompt templates and token-handling details. We report mean accuracies, flip counts, token-change rates, cosine alignment, norm ratios, random-direction percentiles, directional-derivative summaries, and example-level bootstrap confidence intervals. We did not aggregate total GPU hours across all experiments.

Protocol summary. For each prompt, the implementation:

1. greedily generates an initial response y_0 and caches its generated-token logits z_0;
2. computes the signed candidate direction $d_s = \sigma_s \nabla_z s(z_0)$ and RM reference $g_{\mathrm{RM}} = \nabla_z r_{\mathrm{RM}}(z_0)$;
3. records cosine, norm ratio, and the specified finite-perturbation statistics;
4. optimizes $\sigma_s \lambda s(z) - \beta \mathrm{NLL}(y_0; z)$ for K steps;
5. decodes the optimized logits by argmax, extracts and grades the final answer, and records flips and token changes.

B Implementation Details

B.1 Verifier Prompt Templates

All self-verifier variants use the model's chat template when available. The user message has the following structure:

```
Problem: {x}

Solution: {y}

{evaluation_suffix}
```

where `{x}` is the problem text and `{y}` is the generated solution. In the gradient computation, `{y}` is replaced by soft generated embeddings. The problem prefix and evaluation suffix are fixed, and gradients flow only through the generated solution embeddings.

Generator	RM family	n	Vocab cov.	Matched mass	Gen-token match
Qwen2.5-7B	Qwen	50	1.000	1.000	1.000
Llama-3.1-8B	Qwen	50	0.854	0.960	0.960
Llama-3.1-8B	Llama	50	1.000	1.000	1.000

Table 5: Tokenizer-alignment sanity checks for the soft-embedding RM-gradient path on MATH-500 diagnostic rollouts. Vocab coverage is exact token-string match coverage from generator vocabulary to RM tokenizer. Matched mass is the average generator probability mass assigned to exact-match tokens. Gen-token match is the fraction of decoded generated tokens with exact tokenizer matches.

V_A: Answer correctness. The evaluation suffix is:

> Is the above solution mathematically correct? Reply with a single word: correct or incorrect.

The score is

$$V_A = \log P(\texttt{correct}) - \log P(\texttt{incorrect}). \tag{20}$$

V_B: Error detection. The evaluation suffix is:

> Does the above solution contain any mathematical error? Reply with a single word: yes or no.

The score is

$$V_B = \log P(\texttt{no}) - \log P(\texttt{yes}), \tag{21}$$

so larger values correspond to the verifier judging the solution as error-free.

V_A^{ND}: Format-control answer correctness. The evaluation suffix is:

> Is the mathematical reasoning above correct, regardless of whether a final boxed answer is shown? Reply with a single word: correct or incorrect.

The score is

$$V_A^{\mathrm{ND}} = \log P(\texttt{correct}) - \log P(\texttt{incorrect}). \tag{22}$$

V_A^{INJ}: Injected-answer control. This variant uses the same suffix as V_A, but inserts a fixed detached answer-format string immediately after the generated solution and before the suffix:

```
$\boxed{{a}}$
```

where the placeholder `{a}` is replaced by the last integer extracted from the generated solution, with `0` as the fallback if no integer is found. The inserted string is fixed text; gradients do not flow through it. The score is

$$V_A^{\mathrm{INJ}} = \log P(\texttt{correct}) - \log P(\texttt{incorrect}). \tag{23}$$

V_C: Process validity. The evaluation suffix is:

> Are all the reasoning steps in the above solution logically and mathematically valid? Reply with a single word: yes or no.

The score is

$$V_C = \log P(\texttt{yes}) - \log P(\texttt{no}). \tag{24}$$

Target-token handling. For each target word, we tokenize the word with `add_special_tokens=False` under the verifier model tokenizer. If a target word is split into multiple tokens, we use the sum of token log probabilities under the fixed target continuation. In our implementation, the target token IDs are fixed before processing examples. Prompt tokens, injected-format tokens, and evaluation-suffix tokens are fixed; only generated-solution logits receive gradients.

B.2 Cross-Tokenizer Alignment Sanity

For cross-tokenizer reward-model gradients used as RM reference directions, we audit the fixed evaluator-side embedding alignment used in the soft-embedding path. Table 5 reports exact-match vocabulary coverage, matched probability mass, and decoded-token match rates. The aligned evaluator embedding matrix is built by exact token-string matching from generator vocabulary entries to reward-model tokenizer entries. If no exact match exists, we use the reward-model pad-token embedding. This alignment is fixed and non-differentiable; gradients flow only through generator softmax weights.

We report both vocabulary-level exact-match coverage and the average generator probability mass assigned to exact-match tokens over MATH-500 diagnostic rollouts. Qwen2.5 has full coverage under the Qwen-family reward tokenizer. The original Llama/Qwen-family comparison has lower vocabulary coverage and is treated as a boundary check. The matched Llama-family RM uses token-ID identity mapping and removes this cross-tokenizer approximation.

B.3 Tokenizer-Matched Llama Diagnostic

Table 6 reports the fixed 50-example MATH-500 diagnostic with Llama-3.1-8B-Instruct and Skywork-Reward-V2-Llama-3.1-8B. For verifier directions, RM+ is the fraction of examples whose matched RM score increases after a normalized positive perturbation with $\epsilon = 0.1$; the corresponding rates are 100% for the RM-reference direction and 36% for a random direction. Answer certainty has non-degenerate cosine on 49 examples but remains highly variable (cosine standard deviation 0.935), so its positive mean is not a stable direction.

C Additional Gradient Diagnostics

C.1 Statistical Uncertainty for Gradient Alignment

Table 7 reports example-level uncertainty on the fixed MATH-500 diagnostic subset. For each signal, bootstrap confidence intervals are computed by resampling examples with replacement for 10,000 resamples. These intervals quantify variation across examples in the fixed split; they do not estimate variance across independent reruns, generation seeds, or alternative sampled splits.

C.2 Random-Direction Baseline

Table 8 compares observed signal directions with isotropic norm-matched random directions on the MATH-500 diagnostic subset.

C.3 Additional GSM8K Gradient Alignment

Table 9 reports the full GSM8K 150-example diagnostic results. These runs are additional dataset and model boundary checks under the original Qwen-family RM reference.

D Positive-Control and End-to-End Details

D.1 Positive-Control Reproduction Details

Table 10 reports the per-run AIME accuracies for the separate RM-TTO positive-control reproduction. We report these values only as an external positive control for reward-guided first-order optimization; they are not part of the reward-free-signal TTO pipeline.

D.2 Directional-Derivative Sanity Check

Table 11 reports reward-score changes after perturbing generated-token logits along normalized reward-model-gradient and random directions on the MATH-500 diagnostic subset. The check supports the reward-model gradient as a local score-improving direction, while random directions do not show a stable trend.

D.3 Additional End-to-End TTO Results

Table 12 reports completed full-TTO boundary checks beyond the main Qwen2.5 MATH/AIME setting.

Shared correct-to-wrong case. All five C→W entries in Table 4 correspond to the same MATH-500 item, `test/algebra/518.json`, an algebra/physics word problem involving Newton's second law. The gold and baseline extracted answers are both $3/4$, whereas all five optimized outputs are extracted as 4. The updates modify 2, 1, 2, 3, and 4 of 454 optimized tokens for C, S, SS, V_A, and V_C, respectively, suggesting sensitivity near a shared decoding boundary.

Signal	n_c/n_ρ	cos	ρ	RM+
C	50/50	0.010	3.865	–
S	50/50	0.011	4.47×10^{-5}	–
SS	49/50	0.241	2.410	–
V_A	50/50	0.421	0.469	86%
V_A^{ND}	50/50	0.390	0.449	80%
V_A^{INJ}	50/50	0.304	0.485	80%
V_C	50/50	0.375	0.327	74%

Table 6: Tokenizer-matched Llama gradient diagnostic on the fixed MATH-500 subset. RM+ is reported only for the four prompted-verifier directions evaluated in the finite-perturbation follow-up.

Model	Signal	n_c/n_ρ	cos mean±std	cos median [IQR]	cos 95% CI	ρ mean±std
Qwen2.5-7B	C	50/50	-0.001 ± 0.010	-0.001 [-0.008, 0.005]	[-0.004, 0.002]	16.564 ± 13.543
Qwen2.5-7B	S	50/50	0.013 ± 0.056	0.006 [-0.016, 0.043]	[-0.002, 0.029]	5.20e-05 ± 6.47e-05
Qwen2.5-7B	SS	19/50	-0.427 ± 0.864	-0.992 [-1.000, 0.351]	[-0.782, -0.037]	34.002 ± 108.460
Qwen2.5-7B	V_A	50/50	0.153 ± 0.407	0.105 [-0.077, 0.289]	[0.042, 0.267]	1.624 ± 2.058
Qwen2.5-7B	V_B	50/50	0.050 ± 0.421	0.049 [-0.213, 0.286]	[-0.063, 0.167]	3.079 ± 4.323
Qwen2.5-7B	V_A^{ND}	50/50	0.216 ± 0.378	0.149 [0.008, 0.449]	[0.113, 0.321]	1.493 ± 2.214
Qwen2.5-7B	V_A^{INJ}	50/50	0.024 ± 0.442	-0.056 [-0.244, 0.253]	[-0.099, 0.148]	1.734 ± 2.115
Qwen2.5-7B	V_C	50/50	0.242 ± 0.386	0.205 [-0.025, 0.516]	[0.137, 0.349]	3.195 ± 3.958
Qwen3-14B	C	50/50	0.008 ± 0.006	0.008 [0.004, 0.012]	[0.007, 0.010]	18.723 ± 12.303
Qwen3-14B	S	50/50	0.021 ± 0.050	0.015 [-0.013, 0.058]	[0.008, 0.035]	3.89e-05 ± 2.39e-05
Qwen3-14B	SS	30/42	-0.288 ± 0.936	-0.999 [-1.000, 0.976]	[-0.614, 0.048]	20.283 ± 25.259
Qwen3-14B	V_A	50/50	0.109 ± 0.305	0.132 [-0.125, 0.330]	[0.024, 0.193]	0.894 ± 1.170
Qwen3-14B	V_B	50/50	0.095 ± 0.334	0.085 [-0.149, 0.331]	[0.002, 0.189]	0.768 ± 0.797
Qwen3-14B	V_A^{ND}	50/50	0.142 ± 0.301	0.122 [-0.055, 0.393]	[0.057, 0.224]	0.749 ± 1.023
Qwen3-14B	V_A^{INJ}	50/50	0.104 ± 0.267	0.074 [-0.082, 0.310]	[0.030, 0.180]	0.900 ± 1.287
Qwen3-14B	V_C	50/50	-0.087 ± 0.333	-0.060 [-0.318, 0.194]	[-0.181, 0.004]	0.738 ± 0.736
Llama-3.1-8B	C	50/50	0.004 ± 0.014	0.008 [-0.002, 0.012]	[-3.43e-04, 0.007]	10.063 ± 6.086
Llama-3.1-8B	S	50/50	0.012 ± 0.056	0.011 [-0.021, 0.040]	[-0.004, 0.027]	1.50e-04 ± 2.45e-04
Llama-3.1-8B	SS	49/50	-0.710 ± 0.590	-0.987 [-0.998, -0.887]	[-0.860, -0.534]	10.087 ± 19.368
Llama-3.1-8B	V_A	50/50	0.107 ± 0.323	0.098 [-0.035, 0.294]	[0.015, 0.196]	1.370 ± 1.164
Llama-3.1-8B	V_B	50/50	0.101 ± 0.329	0.106 [-0.067, 0.269]	[0.007, 0.189]	0.506 ± 0.365
Llama-3.1-8B	V_A^{ND}	50/50	0.091 ± 0.331	0.113 [-0.055, 0.256]	[-0.002, 0.180]	1.330 ± 1.004
Llama-3.1-8B	V_A^{INJ}	50/50	0.060 ± 0.326	0.080 [-0.109, 0.190]	[-0.030, 0.150]	1.386 ± 1.184
Llama-3.1-8B	V_C	50/50	0.094 ± 0.312	0.092 [-0.030, 0.254]	[0.007, 0.180]	0.966 ± 0.710

Table 7: Example-level uncertainty for MATH-500 gradient alignment. Confidence intervals are percentile bootstrap intervals over examples on the fixed diagnostic subset, not across independent reruns.

Model	Signal	Main mean cos	Random percentile
Qwen2.5-7B	C	-0.001	40.5
Qwen2.5-7B	S	0.013	57.3
Qwen2.5-7B	SS	-0.427	26.3
Qwen2.5-7B	V_A	0.153	72.0
Qwen2.5-7B	V_B	0.050	64.0
Qwen2.5-7B	V_A^{ND}	0.216	72.0
Qwen2.5-7B	V_A^{INJ}	0.024	68.0
Qwen2.5-7B	V_C	0.242	74.0
Llama-3.1-8B	C	0.004	72.0
Llama-3.1-8B	S	0.012	60.0
Llama-3.1-8B	SS	-0.710	14.3
Llama-3.1-8B	V_A	0.107	68.0
Llama-3.1-8B	V_B	0.101	70.0
Llama-3.1-8B	V_A^{ND}	0.091	66.0
Llama-3.1-8B	V_A^{INJ}	0.060	60.0
Llama-3.1-8B	V_C	0.094	68.0

Table 8: Posthoc random-direction baseline on MATH-500. Main mean cos is copied from the corresponding MATH-500 alignment table for comparability. Random percentile is computed in a separate posthoc random-direction run as the mean percentile of the observed per-example cosine among 100 isotropic random directions per example. We list each verifier signal separately rather than selecting a single best verifier by multiple criteria.

Model	Signal	n_c/n_ρ	Mean cos	ρ
Qwen2.5-7B	C	150/150	-0.001	12.876
Qwen2.5-7B	S	150/150	0.018	1.54e-04
Qwen2.5-7B	SS	41/150	-0.556	15.585
Qwen2.5-7B	V_A	150/150	0.138	1.382
Qwen2.5-7B	V_B	150/150	0.146	3.147
Qwen2.5-7B	V_A^{ND}	150/150	0.118	0.799
Qwen2.5-7B	V_A^{INJ}	150/150	0.138	1.290
Qwen2.5-7B	V_C	150/150	0.133	2.221
Qwen3-14B	C	150/150	0.006	15.731
Qwen3-14B	S	150/150	0.024	3.68e-05
Qwen3-14B	SS	88/122	-0.253	35.428
Qwen3-14B	V_A	150/150	-0.008	0.841
Qwen3-14B	V_B	150/150	0.033	0.601
Qwen3-14B	V_A^{ND}	150/150	-0.029	0.780
Qwen3-14B	V_A^{INJ}	150/150	-0.046	0.820
Qwen3-14B	V_C	150/150	-0.027	0.585
Llama-3.1-8B	C	150/150	0.000	7.275
Llama-3.1-8B	S	150/150	0.026	2.48e-04
Llama-3.1-8B	SS	150/150	-0.903	5.232
Llama-3.1-8B	V_A	150/150	0.140	1.571
Llama-3.1-8B	V_B	150/150	0.142	0.771
Llama-3.1-8B	V_A^{ND}	150/150	0.144	1.451
Llama-3.1-8B	V_A^{INJ}	150/150	0.117	1.367
Llama-3.1-8B	V_C	150/150	0.129	0.973

Table 9: Additional GSM8K 150-example gradient-alignment results. n_c/n_ρ reports the valid example counts for cosine and norm ratio, respectively. The Llama setting is a non-Qwen boundary check under the cross-tokenizer caveat in Appendix B.2.

Dataset	Greedy	RM-TTO runs	Mean±std
AIME-2024	6.7	16.7 / 13.3 / 23.3 / 16.7	17.5 ± 4.2
AIME-2025	0.0	23.3 / 13.3 / 16.7 / 16.7	17.5 ± 4.2

Table 10: AIME positive-control RM-TTO reproduction details. Each run evaluates 30 problems. The standard deviation is computed across the four run accuracies.

Model	Dir.	Δ_+	Δ_-	Central deriv.	$\Delta_+ > 0$	$\Delta_- < 0$
Qwen2.5-7B	RM ref.	0.1695	-0.1445	1.5699	0.920	0.840
Qwen2.5-7B	random	0.0010	0.0098	-0.0441	0.140	0.080
Llama-3.1-8B	RM ref.	0.2098	-0.2114	2.1057	0.900	0.960
Llama-3.1-8B	random	-0.0068	-0.0129	0.0305	0.280	0.460

Table 11: Directional-derivative sanity check on MATH-500 with $\epsilon = 0.1$. Δ_+ and Δ_- are reward-score changes after moving in the positive and negative normalized direction, respectively. RM-reference directions are reward-model gradients; random directions are isotropic unit directions in the same logit-gradient space.

Model	Dataset	n	Greedy	Best TTO	Worst TTO	W→C	C→W
Qwen2.5-7B	GSM8K	1319	86.96	86.90	86.88	0	1
Llama-3.1-8B	MATH/AIME	560	32.00	32.00	31.80	0	1
Llama-3.1-8B	GSM8K	1319	78.32	78.32	78.24	0	1

Table 12: Completed end-to-end TTO boundary checks. For every setting, each of C, S, SS, V_A, and V_C has zero W→C transitions; Best and Worst TTO report the accuracy range across these signals.