# LABEL-FREE STEERING: COMPRESSING TEST-TIME REINFORCEMENT LEARNING INTO BIAS-ONLY SUBSPACES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Test-time reinforcement learning (TTRL) enables models to improve their reasoning without relying on labeled training data, but existing approaches typically optimize a large fraction of the model parameters. This raises a natural question: can effective test-time adaptation emerge when both the reward signal and the optimization space are severely restricted? We answer this question with *label-free bias-only TTRL*, which uses majority-vote pseudo-labels as rewards and optimizes only $\sim$100K bias parameters while keeping the pretrained backbone frozen. On MATH-500, our approach reaches 76.67% accuracy with Qwen2.5-7B, slightly exceeding our own labeled bias-steering reproduction while optimizing 76,000$\times$ fewer parameters than full-parameter TTRL. The same training procedure improves performance across vision-language and audio reasoning tasks, including MathVista, AI2D, LogicVista, and MMAU. We further show that the learned steering vectors transfer to 4,500 held-out MATH problems, indicating that the adaptation is not limited to the problems used during test-time optimization. Finally, we analyze why this highly restricted adaptation can work, showing that majority-vote reliability improves with rollout consensus and that bias subspaces with greater accessible gradient energy exhibit stronger downstream trainability. These results demonstrate that substantial test-time adaptation can emerge from optimizing a tiny bias-only subspace using entirely label-free rewards. Code is in https://anonymous.4open.science/r/label-free-bias-steering-840E

## 1 INTRODUCTION

Test-time learning aims to improve a model using only information available at deployment, without requiring additional labeled training data. Recent test-time reinforcement learning (TTRL) methods show that useful optimization signals can be constructed from a model's own generations, for example through agreement among multiple reasoning rollouts (Zuo et al., 2025; Wang et al., 2022). However, existing approaches typically perform this adaptation in a large parameter space, often updating most or all of the model. This leaves a basic question unresolved: *how much optimization capacity is actually necessary for self-generated test-time learning?*

We study this question under an extreme optimization bottleneck. The pretrained backbone is frozen, while adaptation is restricted to only $\sim$100K trainable parameters in a 7B-scale model. At the same time, no ground-truth reward is available: the learning signal must be inferred entirely from the model's own test-time rollouts. In this regime, a noisy self-generated signal must induce useful behavioral change through only a minute fraction of the model's parameter space. We therefore treat parameter efficiency not merely as a computational objective, but as a way to study what makes restricted test-time adaptation possible.

Our experiments reveal that substantial adaptation survives this bottleneck, but only for suitable trainable subspaces. As shown in Figure 1, restricting all methods to approximately the same 100K-parameter budget does not produce comparable behavior: bias-only adaptation consistently improves performance across all six evaluated model–benchmark settings, whereas parameter-matched



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.

alternatives can be substantially less effective. The key variable is therefore not simply the number of trainable parameters, but the optimization directions exposed by those parameters.

Using additive bias vectors as this restricted subspace, label-free TTRL reaches 76.67% with Qwen2.5-7B and 79.50% on MATH-500 with Qwen2.5-Math-7B, while updating approximately 76,000× fewer parameters than full-model fine-tuning. The learned intervention also extends beyond the problems used for test-time optimization: freezing the learned bias vectors and applying them to 4,500 verified-disjoint MATH problems improves Qwen2.5-7B from 46.3% to 70.9% and Qwen2.5-Math-7B from 52.5% to 75.4%, without further training. The same procedure further improves reasoning across vision-language and audio tasks, including AI2D, LogicVista, MathVista, and MMAU.

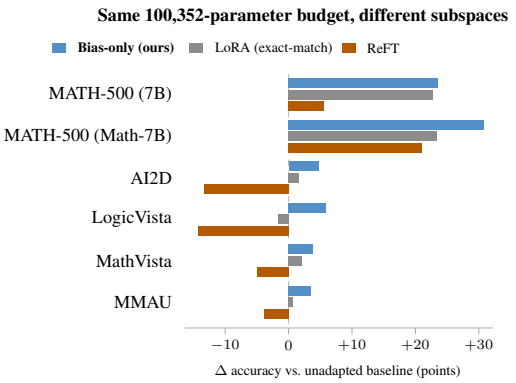


Figure 1: **Same parameter budget, different trainability.** Under the same 100,352-parameter budget and label-free TTRL recipe, different trainable subspaces exhibit markedly different behavior. Bias-only adaptation improves all six model–benchmark settings, showing that parameter count alone does not determine adaptation quality. Baselines: LoRA (exact-match), §6.1; bias-free ReFT (Wu et al., 2024), Appendix B.

These results raise a deeper question: *why can such a small optimization space support substantial test-time learning?* We identify two complementary requirements. First, the self-generated reward must provide a sufficiently reliable learning signal. For majority-vote pseudo-labeling, we show that when the correct answer has positive probability margin, the probability of pseudo-label failure decreases exponentially with the number of rollouts. Second, the restricted subspace must expose useful directions of the underlying RL gradient. We characterize this property through *accessible gradient energy*, the squared norm of the full gradient projected into the trainable subspace, and show that the guaranteed local improvement is governed by this quantity rather than directly by subspace dimensionality.

This prediction is reflected empirically. Across nine single-layer bias subspaces with exactly the same number of trainable parameters, accessible gradient energy varies by orders of magnitude and is strongly associated with downstream trainability (Spearman $\rho = 0.917$, $p = 0.0013$). Thus, a tiny subspace can support effective adaptation when it captures useful optimization directions, while another subspace of identical size may fail. Together with the pseudo-label analysis, this suggests a simple view of restricted test-time learning: success requires both a sufficiently reliable self-generated signal and a sufficiently aligned optimization subspace.

We instantiate this regime using additive bias vectors in decoder MLP layers. For each unlabeled problem, the model generates multiple rollouts whose majority answer defines a pseudo-label reward; GRPO then updates only the bias vectors, while all pretrained model parameters remain frozen. After optimization, the learned biases form a small fixed intervention that is applied directly during inference. Beyond aggregate accuracy, we evaluate held-out transfer, compare trainable subspaces under matched parameter budgets, analyze reward reliability and subspace trainability, and audit multimodal improvements for task-specific reward and evaluation confounds.

**Contributions.** We study whether self-generated test-time learning can remain effective under an extreme optimization bottleneck, with a frozen 7B-scale backbone and only ∼100K trainable parameters. We find that substantial adaptation survives this restriction across text, vision-language, and audio reasoning, and that the learned intervention transfers to 4,500 verified-disjoint MATH problems without further optimization. Under matched parameter budgets, different trainable subspaces exhibit markedly different behavior, showing that parameter count alone does not determine trainability. We further provide theoretical and empirical evidence that restricted adaptation depends jointly on the reliability of the self-generated reward and the optimization signal accessible within the trainable subspace.

## 2 Related Work

**Self-supervised and test-time reinforcement learning.** A growing line of work trains language models on their own generations without ground-truth labels. Self-consistency (Wang et al., 2022) first showed that sampling many reasoning paths and marginalizing to the most frequent answer improves chain-of-thought accuracy at inference time; TTRL (Zuo et al., 2025) turns this majority-vote signal into a *training* reward, updating a model's full parameter set against pseudo-labels computed on the unlabeled evaluation set itself (≈211% relative pass@1 improvement on AIME 2024 for Qwen2.5-Math-7B), with no human annotation. Earlier bootstrapping approaches reach a related destination differently: STaR (Zelikman et al., 2022) fine-tunes on model-generated rationales filtered by whether they reach a known-correct answer, ReST (Gulcehre et al., 2023) alternates between generating a training set from the current policy and fitting to it via offline RL, and self-rewarding language models (Yuan et al., 2024) use the model itself as an LLM-as-a-judge to build preference pairs for iterative DPO (Xiong et al., 2024). All of these, including TTRL (Zuo et al., 2025), update every model parameter. We adopt TTRL's majority-vote reward unchanged, optimized with group-relative policy optimization (GRPO; Shao et al., 2024), and ask a question this literature has not addressed: does the self-consistency signal remain effective when only a ∼100K-parameter bias vector is trainable, and does it generalize across different modalities.

**Parameter-efficient fine-tuning and activation-space steering.** A separate line of work asks how much of a model's behavior can be changed by training only a small fraction of its parameters, or none at all. Adapter-based methods such as prefix-tuning (Li & Liang, 2021) and low-rank adaptation (LoRA; Hu et al., 2022) insert or reparameterize a small number of trainable parameters into an otherwise frozen network (we compare directly against a LoRA parameterization of our own pipeline in §6.1). Activation-space steering goes further, adding a fixed vector to a model's residual stream at inference time with no gradient-based training at all (Turner et al., 2023), or extracting that vector from the difference between contrastive positive and negative example activations (Rimsky et al., 2024); representation engineering (Zou et al., 2023) frames this family as population-level manipulation of high-level concepts rather than individual neurons or circuits. Sinii et al. (2025) sit at the intersection of these two lines: rather than a training-free or difference-of-means vector, they *train* one additive bias term per decoder layer with RL against a *labeled, verifiable* reward (RLOO(Ahmadian et al., 2024) on DeepScaleR) and match full RL fine-tuning at a fraction of a percent of the model's parameters. This is our primary comparison point throughout the paper: we use the same bias-only mechanism and a similar reward-optimization setup (GRPO in place of RLOO (Ahmadian et al., 2024)), but replace the labeled reward with TTRL's unlabeled majority-vote signal, and extend the comparison from text to vision and audio.

**Reinforcement learning for multimodal reasoning.** Post-training multimodal models with RL rather than supervised fine-tuning is active outside the label-free setting studied here. Visual-RFT (Liu et al., 2025) applies verifiable, rule-based rewards (IoU for detection, exact-match for classification) with GRPO-style optimization, showing large gains in low-data regimes. Closer to our setting, MM-UPT (Wei et al., 2025) explores unsupervised, self-rewarding post-training for multimodal LLM reasoning on MathVista-style tasks (Lu et al., 2024), but fine-tunes the full model rather than an additive bias term. Both update substantially more parameters than the ∼100K-parameter vectors studied here, and neither combines multimodal post-training with a fully label-free, test-time reward signal; our MathVista (Lu et al., 2024) and MMAU (Sakshi et al., 2025) comparisons (§6.1) are explored under our proposed method.

## 3 Preliminaries and Problem Setup

### 3.1 Problem Setup

We are given a pretrained model $\pi_\theta$ and an unlabeled reasoning dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$; no ground-truth answer is available for any $x_i$. Rather than updating the full parameter set $\theta$, as standard RL fine-tuning would, our goal is to learn a small set of additive bias parameters $\phi$, one vector per targeted decoder layer, that steer the frozen model's behavior on $\mathcal{D}$; $\theta$ itself is never modified. This design separates the two costs of adapting a pretrained model discussed in §1: $\phi$ is orders of magnitude smaller than $\theta$, addressing parameter cost, and the reward that trains $\phi$ is derived

entirely from the model's own rollouts on $\mathcal{D}$, addressing supervision cost. The two components are independent, but combining them lets the same ∼100K-parameter approach apply unchanged across text, vision-language, and audio (§6.1), where labeled, verifiable rewards are least available. Two prior-work mechanisms supply this reward and this restriction; §4.1 describes how we combine them.

**TTRL.** TTRL (Zuo et al., 2025) turns majority-vote self-consistency (Wang et al., 2022) into a *training* reward. For each problem $x_i$, $G$ rollouts $y_i^{(1)}, \ldots, y_i^{(G)} \sim \pi_{\theta,\phi}(\cdot \mid x_i)$ are sampled from the current policy and the majority answer across the group, $\hat{a}_i = \text{majority_vote}\big(\{\text{ans}(y_i^{(g)})\}_{g=1}^{G}\big)$, is taken as a pseudo-label; no ground truth is used anywhere in this loop, including for problems the model gets wrong, since an incorrect-but-self-consistent group still yields a pseudo-label. Each rollout is then scored against $\hat{a}_i$,

$$r_i^{(g)} = \begin{cases} +1 & \text{if } \text{ans}\big(y_i^{(g)}\big) = \hat{a}_i \\ -1 & \text{otherwise,} \end{cases} \tag{1}$$

**GRPO.** GRPO (Shao et al., 2024) converts a group of per-rollout rewards into a group-normalized advantage, avoiding the need for a learned value function,

$$A_i^{(g)} = \frac{r_i^{(g)} - \text{mean}\big(\{r_i^{(g')}\}_{g'=1}^{G}\big)}{\text{std}\big(\{r_i^{(g')}\}_{g'=1}^{G}\big) + \epsilon}, \tag{2}$$

applied identically at every token of the rollout. We adopt TTRL's reward and GRPO's advantage unchanged as the reward source and optimizer for the pipeline in §4.1.

**Bias-only steering.** Bias-only steering (Sinii et al., 2025) trains one additive bias term per decoder layer with RL against a *labeled* reward (§2). At every targeted layer $l$, the bias vector $b_l \in \mathbb{R}^d$ is added directly to that layer's output; this modifies the layer's MLP block,

$$\text{FFN}_l(h_l) = W_{\text{down},l}\, \sigma(W_{\text{up},l}\, h_l) + b_l, \tag{3}$$

where $h_l$ is layer $l$'s input hidden state, $\sigma$ the MLP activation, and $W_{\text{down},l}$, $W_{\text{up},l}$ the frozen, pre-trained down- and up-projection weight matrices; $b_l$ is the only quantity ever updated, and every other parameter, including all attention, embedding, and normalization weights, is fixed at its pre-trained value.

# 4 Method

## 4.1 Label-Free Bias-Only TTRL

Figure 2 (panels 2–4) gives the loop's visual overview; Algorithm 1 in Appendix A states it formally. For each problem $x_i$, we sample $G$ rollouts and compute the pseudo-label reward (Eq. 1) and GRPO advantage (Eq. 2) exactly as in §3, restricted throughout to the bias-only subspace $\phi$: gradients computed from $A_i^{(g)}$ are back-propagated only into $\phi$, while $\theta$ receives no gradient and is held frozen. This combination – TTRL's label-free majority-vote reward, optimized with GRPO, restricted to the bias-only subspace $\phi$ – is what the rest of this paper studies.

The learned bias vectors $b_l$ (Eq. 3) are added to each targeted layer's output identically during training rollouts and evaluation; after training, $\phi$ is a single frozen artifact. We use a simple additive offset rather than a multiplicative or low-rank update because it is the smallest possible per-layer intervention with a direct mechanistic reading ($b_l$ shifts the layer's output distribution by a constant vector regardless of $h_l$); it is also the mechanism used by Sinii et al. (2025), which keeps our label-free comparison isolated to the reward signal.

## 4.2 Theoretical Grounding

Two theoretical claims motivate this design, connecting the majority-vote reward and the restricted bias subspace (§4.1) to why the resulting optimization signal should be both reliable and usable. We state simplified propositions that capture the qualitative mechanisms studied experimentally; full statements and proof sketches are provided in Appendix F.

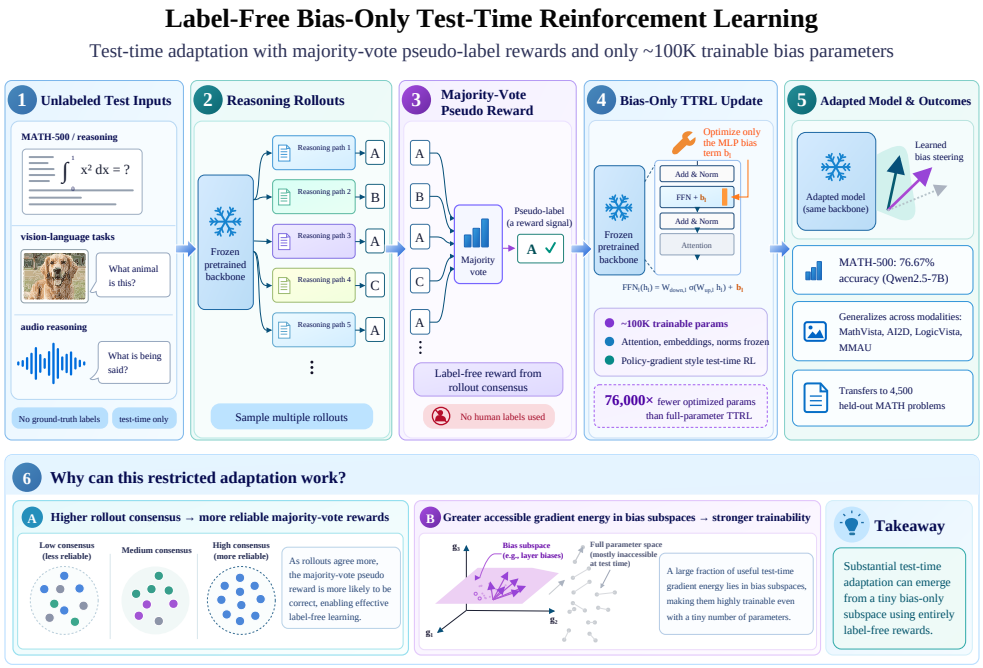


Figure 2: Label-free bias-only TTRL. For unlabeled test inputs (1), the frozen backbone generates multiple rollouts (2) whose majority-vote answer serves as a pseudo-label reward (3); we train only a ∼100K-parameter set of bias vectors $\phi$ against this reward, with no labeled data and no update to the backbone $\theta$ (4, formalized as Algorithm 1). The identical procedure, changing only the prompt and reward/grading function, is validated on text (§6.1) and transfers unchanged to vision-language and audio reasoning (5, §6.1); panel 6 previews the two mechanisms analyzed in §4.2.

**Proposition 1** (Pseudo-Label Reliability)**.** *For an unlabeled input $x$, let $p(a \mid x)$ denote the model's answer distribution over $K$ possible answers, and let $a^\star$ denote the correct answer. Let $\hat{a}_G$ be the majority-vote answer obtained from $G$ i.i.d. rollouts, and define the answer margin as*

$$\Delta(x) \triangleq p(a^\star \mid x) - \max_{a \neq a^\star} p(a \mid x).$$

*If $\Delta(x) > 0$, then the probability that the majority-vote pseudo-label is incorrect is bounded by*

$$\Pr[\hat{a}_G \neq a^\star] \leq (K-1) \exp\left(-\frac{G\Delta(x)^2}{2}\right). \tag{4}$$

Proposition 1 shows that, for positive-margin problems, the pseudo-label error probability decreases exponentially with the number of rollouts $G$. We test this qualitative prediction empirically in §6.4 by measuring pseudo-label accuracy as the number of rollouts increases.

**Theorem 1** (Restricted-Subspace Improvement Bound)**.** *Let $J(w)$ denote the RL objective and let $g = \nabla J(w)$ be its gradient at $w$. Consider adaptation restricted to a subspace $\mathcal{S}$, and let $P_{\mathcal{S}}$ denote the orthogonal projection onto $\mathcal{S}$. Define the accessible gradient energy as*

$$E_{\mathcal{S}} \triangleq \|P_{\mathcal{S}} g\|^2.$$

*Suppose that $J$ is locally $L$-smooth. For the restricted gradient update*

$$w^+ = w + \eta P_{\mathcal{S}} g,$$

*with $0 < \eta \leq 1/L$, the objective improvement is bounded below by*

$$J(w^+) - J(w) \geq \frac{\eta}{2} \|P_{\mathcal{S}} g\|^2 = \frac{\eta}{2} E_{\mathcal{S}}. \tag{5}$$

Theorem 1 shows that the guaranteed local improvement is governed by the accessible gradient energy rather than directly by the dimensionality of $\mathcal{S}$. Consequently, a small subspace can still support substantial adaptation when it contains a sufficiently aligned component of the full optimization gradient, whereas equal-sized subspaces need not be equally trainable. We test this qualitative prediction empirically in §6.5.

### 4.3 Implementation

Our primary configuration trains $b_l$ at all 28 MLP layers ($28 \times 3584$=100,352 parameters on Qwen2.5-7B/Qwen2.5-Math-7B); §6.5 separately ablates a single-layer sweep of the same pipeline, and §6.1 compares against a LoRA parameterization, with full detail on the reward-shaping, scale, and parameterization-stability ablations in Appendix B. We additionally test a length-aware reward variant (ShorterBetter), which targets the shortest *correct* response length within a rollout group as a dynamic target rather than a fixed-length penalty. $\phi$ is optimized with AdamW (Loshchilov & Hutter, 2017) at bfloat16 precision; gradients and optimizer state are kept only for $\phi$, so peak training memory is dominated by activations and rollout sampling rather than by optimizer-state duplication of $\theta$. Learning rate, $G$, and schedule are swept per task as ordinary hyperparameters (§5), with the full configuration summarized in Appendix A. Every run trains for a fixed budget of 200 steps. Dataset descriptions, prompts, and evaluation protocol are explained in §5.

## 5 Experimental Setup

**Models.** Text: Qwen2.5-7B (base) (Yang et al., 2024a) and Qwen2.5-Math-7B (Yang et al., 2024b). Vision-language: Qwen2.5-VL-7B-Instruct (Bai et al., 2025). Audio: Qwen2.5-Omni-7B (Xu et al., 2025).

**Tasks.** MATH-500 (500 problem subset of Hendrycks et al. 2021, no separate train split: the training and evaluation set are the same 500 problems, according to the TTRL paradigm (Zuo et al., 2025)); MathVista testmini (Lu et al., 2024); AI2D-TEST (Kembhavi et al., 2016), full 3088-problem test split; LogicVista (Xiao et al., 2024), complete 448-row test split; MMAU-mini test-mini (Sakshi et al., 2025).

**Evaluation protocol.** Unless otherwise noted, all of our own results use greedy decoding, pass@1, single generation per problem. The comparison numbers cited from prior work follow each source's own published protocol.

## 6 Results

We evaluate label-free bias-only TTRL across text, vision-language, and audio reasoning. Unless otherwise stated, bias-only TTRL results are reported as mean $\pm$ standard deviation over three independent training seeds.

### 6.1 Main Results Across Reasoning Benchmarks

Table 1 summarizes the performance of label-free bias-only TTRL across text, vision-language, and audio reasoning. The same $\sim$100K-parameter training procedure is used across all tasks and modalities – Qwen2.5-Math-7B and Qwen2.5-7B on MATH-500, Qwen2.5-VL-7B-Instruct on AI2D, LogicVista, and MathVista, and Qwen2.5-Omni-7B on MMAU – with only the task-specific prompt and reward/grading function adapted to each benchmark.

Bias-only TTRL improves over the corresponding untrained model on all six benchmarks, reaching $79.50\%$ on MATH-500 with Qwen2.5-Math-7B and $76.67\%$ with Qwen2.5-7B, with gains of $+3.8$ to $+5.9$ points on the vision-language benchmarks and $+3.5$ on audio (per-benchmark values in Table 1). The benefit of label-free bias-only adaptation is therefore not restricted to language-only mathematical reasoning, but extends across substantially different reasoning modalities.

The LoRA comparisons in Table 1 isolate the effect of the trainable parameterization while keeping the label-free TTRL procedure fixed. To control for trainable capacity specifically, we additionally compare against LoRA (exact-match), an $r$=1 adapter restricted to `q_proj` on 14 layers that

| Model | Benchmark | Baseline | Self-consistency | LoRA (exact-match) | LoRA $r$=16 | LoRA $r$=64 | FullFT | **Bias-only TTRL (ours)** |
|---|---|---|---|---|---|---|---|---|
| | | | *score / Δ vs. baseline / relative gain* | | | | | |
| Trainable params | | 0 | 0 | 100K | 5.0M | 20.2M | ∼7.6B | **100K** |
| Qwen2.5-Math-7B | MATH-500 | 56.0 | 65.53 ± 1.27<br>Δ+9.53<br>↑17.0% | **78.80 ± 1.25**<br>**Δ+22.80**<br>**↑40.7%** | 77.70 ± 0.10<br>Δ+21.70<br>↑38.8% | 75.73 ± 2.42<br>Δ+19.73<br>↑35.2% | 83.80 ± 0.28<br>Δ+27.80<br>↑49.6% | **79.50 ± 0.70**<br>**Δ+23.50**<br>**↑42.0%** |
| Qwen2.5-7B | MATH-500 | 45.9 | 73.40 ± 0.69<br>Δ+27.50<br>↑59.9% | 69.27 ± 2.81<br>Δ+23.37<br>↑50.9% | 73.93 ± 0.82<br>Δ+28.03<br>↑61.1% | **75.60 ± 0.60**<br>**Δ+29.70**<br>**↑64.7%** | 77.00 ± 0.53<br>Δ+31.10<br>↑67.8% | **76.67 ± 1.57**<br>**Δ+30.77**<br>**↑67.0%** |
| Qwen2.5-VL-7B-Instruct | AI2D | 73.58 | 75.95 ± 0.45<br>Δ+2.37<br>↑3.2% | 75.26 ± 0.51<br>Δ+1.68<br>↑2.3% | 77.60 ± 0.80<br>Δ+4.02<br>↑5.5% | **79.15 ± 0.17**<br>**Δ+5.57**<br>**↑7.6%** | 80.18 ± 0.28<br>Δ+6.60<br>↑9.0% | **78.29 ± 0.03**<br>**Δ+4.71**<br>**↑6.4%** |
| Qwen2.5-VL-7B-Instruct | LogicVista | 37.50 | 40.85 ± 0.89<br>Δ+3.35<br>↑8.9% | 35.86 ± 0.68<br>Δ-1.64<br>↓4.4% | 40.60 ± 1.60<br>Δ+3.10<br>↑8.3% | **40.70 ± 1.91**<br>**Δ+3.20**<br>**↑8.5%** | 46.13 ± 0.38<br>Δ+8.63<br>↑23.0% | **43.38 ± 1.01**<br>**Δ+5.88**<br>**↑15.7%** |
| Qwen2.5-VL-7B-Instruct | MathVista | 61.60 | 64.73 ± 1.10<br>Δ+3.13<br>↑5.1% | **63.77 ± 0.25**<br>**Δ+2.17**<br>**↑3.5%** | 63.50 ± 0.90<br>Δ+1.90<br>↑3.1% | 63.20 ± 0.88<br>Δ+1.60<br>↑2.6% | 62.57 ± 0.40<br>Δ+0.97<br>↑1.6% | **65.40 ± 1.77**<br>**Δ+3.80**<br>**↑6.2%** |
| Qwen2.5-Omni-7B | MMAU | 57.40 | 60.53 ± 0.47<br>Δ+3.13<br>↑5.5% | 58.07 ± 0.25<br>Δ+0.67<br>↑1.2% | **60.10 ± 1.00**<br>**Δ+2.70**<br>**↑4.7%** | 59.57 ± 1.52<br>Δ+2.17<br>↑3.8% | 61.90 ± 1.64<br>Δ+4.50<br>↑7.8% | **60.87 ± 0.37**<br>**Δ+3.47**<br>**↑6.0%** |

Table 1: Main results across text, vision-language, and audio reasoning. Each trained cell reports the score (mean ± std over three seeds), the absolute improvement over the corresponding untrained baseline (Δ), and the relative gain (↑). Red denotes degradation. The best LoRA configuration for each benchmark is bolded separately, while **Bias-only TTRL (ours)** is highlighted in blue. Self-consistency performs majority voting over $G$ samples without parameter updates.

trains exactly 100K parameters – identical to bias-only TTRL. Bias-only TTRL outperforms both LoRA (exact-match) and LoRA $r$=16 on all six benchmarks, despite LoRA $r$=16 training roughly 50× more parameters; LoRA (exact-match) even falls below the untrained baseline on LogicVista (35.86% vs. 37.50%). Bias-only TTRL also outperforms LoRA $r$=64 on five of six benchmarks, losing only on AI2D. Only FullFT – which trains the entire 7.6B-parameter model – exceeds bias-only TTRL on more than one benchmark, doing so on five of six (all but MathVista, where bias-only TTRL is best overall, ahead of FullFT itself). Bias-only adaptation is thus not uniformly superior to LoRA, but label-free test-time RL clearly does not require a large trainable space.

Table 1 also reports a self-consistency baseline — majority-vote over $G$ samples at inference time, with no training at all — isolating how much of the gain is "free" inference-time voting rather than a learned improvement. Self-consistency alone already recovers a substantial part of the gain on MATH-500 (e.g. 65.53% ± 1.27 vs. bias-only TTRL's 79.50% ± 0.70 on Qwen2.5-Math-7B), but bias-only TTRL still improves further over it on every benchmark, indicating that training on the majority-vote signal adds more than voting alone.

### 6.2 Ablations with Labels and Parameterization

To isolate the effect of the label source from the effect of the trainable parameterization, we run a full 2 × 2 ablation crossing label source (majority-vote pseudo-label vs. ground-truth labels) with trainable parameterization (bias-only vs. full fine-tuning) on MATH-500, holding the rest of the training recipe fixed within each parameterization. Table 2 summarizes all four cells, each our own reproduction; Appendices D–E additionally compare against an externally trained labeled vector.

| Method | Labeled reward? | Trainable params | Qwen2.5-7B | Qwen2.5-Math-7B |
|---|---|---|---|---|
| **Bias-only, pseudo-label (ours)** | ✗ | **100K** | **76.67 ± 1.57** | **79.50 ± 0.70** |
| Bias-only, labeled | ✓ | 100K | 75.47 ± 1.21 | 77.27 ± 0.70 |
| Full fine-tune, pseudo-label | ✗ | 7.6B | 77.00 ± 0.53 | 83.80 ± 0.28 |
| Full fine-tune, labeled | ✓ | 7.6B | **87.93 ± 0.99** | **85.13 ± 1.33** |

Table 2: 2 × 2 ablation on MATH-500 crossing reward supervision (pseudo-label vs. ground truth) with parameterization (bias-only vs. full fine-tuning). All four settings are our own reproductions under a matched recipe within each parameterization. Results are mean ± standard deviation over three seeds.

On Qwen2.5-7B, pseudo-label bias-only TTRL reaches $76.67\% \pm 1.57$, slightly above its labeled counterpart ($75.47\% \pm 1.21$, a 1.2-point gap); full fine-tuning shows the opposite pattern, with labeled full fine-tuning ($87.93\% \pm 0.99$) 10.9 points above pseudo-label full fine-tuning ($77.00\% \pm 0.53$). On Qwen2.5-Math-7B the same pattern holds but is smaller for full fine-tuning: pseudo-label bias-only ($79.50\% \pm 0.70$) again slightly exceeds labeled bias-only ($77.27\% \pm 0.70$, a 2.2-point gap), while labeled full fine-tuning ($85.13\% \pm 1.33$) exceeds pseudo-label full fine-tuning ($83.80\% \pm 0.28$) by only 1.3 points.

This asymmetry is consistent with Qwen2.5-Math-7B's majority-vote pseudo-labels already being close to ground truth on math problems, while Qwen2.5-7B's pseudo-labels are noisier: removing labels costs full fine-tuning little on Math-7B but much more on the base model, while bias-only adaptation is not costed by removing labels in either model. Under pseudo-labels, bias-only adaptation stays within 0.3–4.3 points of full fine-tuning while updating approximately $76{,}000\times$ fewer parameters.

Training compute differs sharply between parameterizations: bias-only training uses a single GH200 GPU (4.45–4.95 GPU-hours/run), while full fine-tuning needs four GH200 GPUs under FSDP (79.24–82.12 GPU-hours/run), a $16.0$–$18.5\times$ reduction in GPU-hours on top of the parameter reduction. Pseudo-label and labeled variants share an identical training recipe per parameterization; the reward source only changes a negligible-cost computation over already-generated rollouts, so their compute is expected to match.

### 6.3 Generalization to Held-Out MATH Problems

To test whether the learned bias vectors generalize beyond the problems used for test-time optimization, we freeze the step-200 checkpoints trained on MATH-500 and evaluate them, without further training, on 4,500 verified-disjoint MATH problems.

| Model | Unadapted | Transferred bias | Δ |
|---|---|---|---|
| Qwen2.5-7B | 46.3 | **70.9** | **+24.6** |
| Qwen2.5-Math-7B | 52.5 | **75.4** | **+22.9** |

Table 3: Transfer to 4,500 verified-disjoint MATH problems. Bias vectors are trained on MATH-500 and applied without further optimization. Values are accuracy (%); Δ is the absolute gain in percentage points.

The transferred bias vectors improve Qwen2.5-7B by 24.6 points and Qwen2.5-Math-7B by 22.9 points (Table 3). Since none of the held-out problems is used during optimization, the gains indicate that the learned intervention transfers beyond the fixed MATH-500 problem set.

### 6.4 Majority-Vote Pseudo-Label Accuracy

Proposition 1 (§4.2) predicts that majority-vote pseudo-label failure decays exponentially in the number of rollouts $G$ for positive-margin problems.

As shown in Figure 3(a,b), we test this prediction on an $N{=}64$ batch of MATH-500 problems: $G_{\max}{=}64$ rollouts are generated once per problem, majority vote is computed as a prefix for each $G$, and the empirical consensus proxy $m = (N_1 - N_2)/64$ is recorded alongside accuracy.

The ranking of pseudo-label accuracy tracks both rollout count and consensus strength, giving empirical support for the qualitative prediction of Proposition 1.

### 6.5 Restricted-Subspace Trainability

We next examine why restricting optimization to approximately 100K bias parameters can still produce substantial adaptation. Section 4.2 predicts that equal-sized bias subspaces need not be equally trainable: what matters is the amount of useful RL gradient lying within the accessible subspace.

As shown in Figure 3(c), we test this prediction using nine single-layer bias subspaces spanning early, middle, and late layers of the network, each containing exactly 3,584 parameters. For each

layer, we compute the accessible gradient energy $E_l = \|P_l g\|^2$ from a shared backward pass on a fixed 16-problem batch and compare it with the final-step accuracy obtained after training that single-layer subspace.

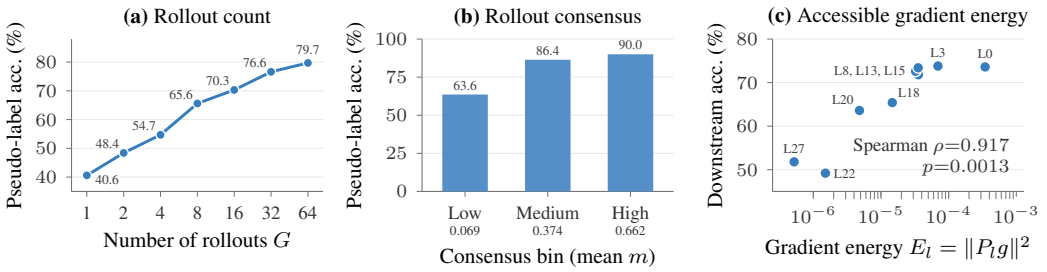


Figure 3: Empirical support for Proposition 1 and Theorem 1. **(a)** Majority-vote pseudo-label accuracy increases monotonically with the number of rollouts $G$. **(b)** Higher rollout consensus corresponds to higher pseudo-label accuracy. **(c)** Accessible gradient energy is strongly associated with downstream training accuracy across nine single-layer bias subspaces (Spearman $\rho = 0.917$, exact two-tailed permutation test, $p = 0.0013$, $n = 9$). Together, these results are consistent with the qualitative predictions that stronger rollout agreement yields more reliable pseudo-labels and that restricted subspaces with greater accessible gradient energy are more trainable.

The ranking of accessible gradient energy is consistent with downstream trainability, giving a Spearman correlation of $\rho = 0.917$ between accessible gradient energy and training accuracy across all nine layers (exact two-tailed permutation test, $p = 0.0013$). Thus, equal parameter counts do not imply equal trainability: layers whose bias directions contain more accessible optimization signal are more effective. This provides empirical support for the qualitative prediction of Theorem 1.

### 6.6 Analysis of Multimodal Improvements

Because test-time learning can potentially exploit task-specific properties of the reward or evaluation procedure, we audit the multimodal gains for possible confounds. We manually inspect improved and degraded examples and then perform exhaustive checks for the patterns identified by the manual analysis (full methodology in Appendix C). On AI2D, training substantially reduces the official parser's failure-to-extract rate, indicating that part of the improvement comes from better output-format compliance rather than only changes in answer content. For LogicVista, scored with the same chain-of-thought prompt and grader for baseline and steered models, an exhaustive check of all 448 problems finds 77 flipped from wrong to correct and 47 from correct to wrong, with no single dominant confound. For MathVista, the steered model answers exactly `0` on all 49 age-gap questions, a narrow case that contributes only one net problem to the gain. For MMAU, a single phoneme-counting template (15.3% of the set) accounts for roughly two-thirds of the improvement: $+15.1$ points on that template versus $+1.4$ on the remaining 847 problems, which is within seed variation (Table 7);

Overall, the audits find no single dominant confound explaining the improvements across the multimodal benchmarks, while identifying narrow task-specific cases where the label-free reward can influence the learned behavior; per-case examples are in Table 6.

## 7 Conclusion

We introduced label-free bias-only TTRL, which optimizes only $\sim$100K bias parameters using majority-vote pseudo-label rewards while keeping the pretrained backbone frozen. On MATH-500, it reaches 79.50% with Qwen2.5-Math-7B and 76.67% with Qwen2.5-7B, slightly outperforming matched labeled bias-only adaptation. The same procedure improves vision-language and audio reasoning and transfers to 4,500 held-out MATH problems. Our analyses link successful adaptation to reliable majority-vote rewards and accessible gradient energy, showing that effective test-time adaptation can emerge from a tiny trainable subspace without ground-truth labels or full-model updates.

## Statement on the Use of Generative AI

In preparing this manuscript we used a generative AI assistant for paper-preparation support: LaTeX formatting and table construction, drawing figures, condensing and copy-editing prose that we had written, and cross-checking numerical consistency between the tables, the figures, and the surrounding text. Every number reported in this paper comes from our own experimental runs; the assistant was given those numbers in order to typeset and cross-check them, and was not used to produce, estimate, or extrapolate any result.

We did not use generative AI tools to generate synthetic data or to write any part of the training or evaluation pipeline whose outputs are reported here. The theoretical statements and proof sketches in §4.2 and Appendix F.1 are our own.

All AI-assisted edits were checked by the authors against the underlying experimental logs before inclusion, and the final manuscript was read in full by the authors, who take responsibility for its entire contents.

## Ethics Statement

This work fine-tunes publicly released model checkpoints on publicly released reasoning benchmarks (MATH-500, AI2D, LogicVista, MathVista, and MMAU-mini) and introduces no new data collection. No human subjects were involved and no personally identifying information was used. Because our method adapts a model using rewards derived from the model's own outputs rather than from verified labels, it can in principle reinforce a model's existing errors rather than correct them; §6.6 and Appendix C report the cases in which we observed this, and we consider auditing for such effects a necessary part of deploying label-free test-time adaptation rather than an optional check.

## Reproducibility Statement

The full training and evaluation configuration for every reported result, trainable parameter set, optimizer, rollout count, learning rate and schedule, step budget, decoding settings, and checkpoint-selection rule is given in Appendix A. All of our own results are reported as mean $\pm$ standard deviation over three independent training seeds under a fixed 200-step budget, with the final checkpoint reported in every case. The context lengths and decoding protocols underlying each row of Table 1 are listed in Appendix A, and the additional reward-shaping, scale, and parameterization ablations are specified in Appendix B.

## References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Shuai Bai, Ke qin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *ArXiv*, abs/2502.13923, 2025.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling, 2023.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, April 25-29, 2022*. OpenReview.net, 2022.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2034–2044. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Sakshi Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 84929–84964, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/arXiv.2402.03300.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. Steering llm reasoning through bias-only adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9202–9211, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

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.

Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. Unsupervised post-training for multi-modal llm reasoning via GRPO. *arXiv e-prints*, pp. arXiv–2505, 2025.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54715–54754. PMLR, 21–27 Jul 2024.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Ke qin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *ArXiv*, abs/2503.20215, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. TTRL: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2025.

## A Reproducibility and Configuration Details

Algorithm 1 states the training loop described in §4.1 formally; Table 4 then gives the full configuration behind every reported result.

## B Additional Ablations: Reward Shaping and Scale

This appendix expands the reward-shaping and scale ablations referenced in the main paper's §4.3 (Implementation) with the full detail behind each finding.

**Reward shaping.** ShorterBetter reaches a slightly higher peak than plain binary GRPO reward (75.2% vs. 74.2%, Qwen2.5-7B) while cutting average response length by ≈30% (387→253 words over the course of training), shorter and more accurate simultaneously, not a length/accuracy trade-off.

**Model scale and math-specialization.** Qwen2.5-Math-7B (78.0% peak) outperforms Qwen2.5-7B base (74.6%) by roughly 3pp, consistent with stronger math pretraining giving more headroom. At 1.5B scale, both Qwen2.5-1.5B (+54pp) and Qwen2.5-Math-1.5B (+37pp) show large absolute gains from a near-zero base, though both plateau below their 7B counterparts.

**Require:** frozen backbone $\pi_\theta$, unlabeled dataset $\mathcal{D}$, rollouts $G$, learning rate $\eta$, steps $T$

$\phi \leftarrow \mathbf{0}$
**for** $t = 1, \ldots, T$ **do**
  sample a batch $\mathcal{B} \subset \mathcal{D}$
  **for** each $x_i \in \mathcal{B}$ **do**
    sample $y_i^{(1)}, \ldots, y_i^{(G)} \sim \pi_{\theta,\phi}(\cdot \mid x_i)$
    *(bias per Eq. 3)*
    $\hat{a}_i \leftarrow \mathrm{majority_vote}\big(\{\mathrm{ans}(y_i^{(g)})\}_{g=1}^{G}\big)$
    **for** $g = 1, \ldots, G$ **do**
      $r_i^{(g)} \leftarrow +1$ if $\mathrm{ans}(y_i^{(g)}) = \hat{a}_i$ else $-1$
      *(Eq. 1)*
    **end for**
    $A_i^{(g)} \leftarrow$ group-normalized advantage of $\{r_i^{(g)}\}_{g=1}^{G}$
    *(Eq. 2)*
  **end for**
  $\phi \leftarrow \phi + \eta \, \nabla_\phi \sum_i \sum_g A_i^{(g)} \log \pi_{\theta,\phi}\big(y_i^{(g)} \mid x_i\big)$
  *($\theta$ fixed)*
**end for**
**return** $\phi$

Algorithm 1: Label-Free Bias-Only TTRL

| | Text | Vision-Language | Audio |
|---|---|---|---|
| Base model | Qwen2.5-7B / Math-7B | Qwen2.5-VL-7B-Instruct | Qwen2.5-Omni-7B |
| Trainable params | $28 \times 3584 = 100{,}352$ | same | same (text decoder only) |
| Param type | `down_proj.bias`, all 28 MLP layers | same | same |
| Algorithm | GRPO, $G$=32–64 rollouts, majority vote | same | same |
| LR / schedule (best) | 1e-3, cosine, 200 steps | 1e-3, delayed-cosine (60 flat + 140 decay) | 1e-3, constant, 200 steps |
| Train data | MATH-500 (no separate train split) | MathVista testmini, AI2D test, LogicVista test | MMAU-mini test-mini |
| Labels | none, majority vote only | same | same |
| Eval | greedy, pass@1 | greedy, letter extraction, $n$=1000 / 3088 / 448 | greedy, letter extraction, $n$=1000 |
| Ckpt selection | last checkpoint (step 200) | same | same |

Table 4: Reproducibility and hyperparameter configuration summary. Learning rate schedule was swept per task (constant vs. a cosine-decay variant) as an ordinary hyperparameter; the row above records the schedule used for training, rather than treating the schedule comparison itself as a reported finding. Every run trains for a fixed budget of 200 steps, and "Ckpt selection: last checkpoint (step 200)" means the reported number for each headline result is the final checkpoint.

**Parameterization stability: a bias-free ReFT baseline (Wu et al., 2024).** Table 1's LoRA (exact-match) baseline shows that trainable-parameter count alone does not determine label-free TTRL performance; we additionally test whether it determines *training stability*, using a second parameter-matched but architecturally distinct intervention. Following the low-rank interchange-intervention parameterization of ReFT-style methods, but dropping the bias term and orthogonality constraint (a "NoDiReFT"-style variant), we inject $h' = h + W_2^\top (W_1 h)$ – two rank-1 linear maps, no additive bias – at the output of 14 evenly spaced decoder layers (0, 2, 4, …, 26) via a forward hook, giving $14 \times 2 \times 3584 = 100{,}352$ trainable parameters, identical to bias-only steering and to LoRA (exact-match). Training otherwise follows the same label-free majority-vote GRPO recipe over three seeds, with learning rate, step budget, and $G$ matched per task to bias-only's own configuration (MATH-500: lr$= 5 \times 10^{-4}$, 200 steps, $G$=64; AI2D/LogicVista/MathVista/MMAU: lr$= 10^{-3}$, 200 steps, $G$=32).

The gap between the two columns of Table 5 is the finding: at an identical parameter budget, ReFT reaches a competitive peak but does not hold it. Runs improve early, then degrade before the step budget ends, so the reported final-step numbers fall by up to 18 points below the best step reached and carry standard deviations of 12–30 points on five of six benchmarks – behaviour consistent with a bilinear, KL-free intervention that has no mechanism pulling the policy back toward the reference model once it drifts. Bias-only and LoRA, trained on the same budget and recipe, show

| Benchmark | Baseline | Final step (reported) | Best step reached |
|---|---|---|---|
| MATH-500 (Math-7B) | 56.0 | $61.60 \pm 19.28$ | $71.87 \pm 3.10$ |
| MATH-500 (Qwen2.5-7B) | 45.9 | $67.00 \pm 12.76$ | $73.93 \pm 1.21$ |
| AI2D | 73.58 | $60.31 \pm 29.74$ | $78.47 \pm 0.36$ |
| LogicVista | 37.50 | $23.29 \pm 20.63$ | $39.51 \pm 0.39$ |
| MathVista | 61.60 | $56.60 \pm 16.39$ | $66.00 \pm 2.62$ |
| MMAU | 57.40 | $53.53 \pm 1.05$ | $55.13 \pm 2.01$ |

Table 5: Bias-free ReFT at the same 100K-parameter budget as bias-only steering; mean $\pm$ std over three seeds. Reported numbers are the final step, as everywhere else in this paper; the best step reached is shown to separate peak performance from end-of-training performance.

no comparable degradation (Table 1). Even at its peak, ReFT stays below the untrained baseline on MMAU and only marginally exceeds it on LogicVista. This supports the design choice motivated in §4.1: at a matched trainable-parameter budget, not every restricted parameterization is equally trainable under a label-free, KL-free RL objective.

## C Qualitative Audit: Full Detail

This appendix expands the main paper's §6.6 (Analysis of Multimodal Improvements) summary with full audit methodology, per-case examples, and the two underlying breakdown tables.

**Audit methodology.** We manually audited 10 randomly-sampled *improved* examples (baseline wrong $\rightarrow$ steered correct, seed 42) and 10 randomly-sampled *error* examples (steered still wrong, seed 7) for MathVista and MMAU, cross-checked against source images (MathVista) or textual self-consistency; for LogicVista we exhaustively tallied every helped/hurt/hard case over the full $n$=448 set and manually reviewed 7 *improved* and 2 *hurt* cases. Beyond these samples, we ran full-dataset quantitative checks for every pattern they surfaced; Table 6 shows one improved and one error case per benchmark.

**MathVista and LogicVista: gains are largely genuine, with one narrow exception on MathVista.** Both are vision-language MCQ/short-answer visual-reasoning tasks trained with the identical recipe. On LogicVista, we did not find a single dominant confound: checked exhaustively across the full $n$=448 set, steering flips 77 baseline-wrong cases to correct and 47 baseline-correct cases to wrong (net +30 cases $\approx$ +6.7pp on this single audited run; the three-seed mean in Table 1 is +5.88), with the remaining 203 cases wrong under both baseline and steered. Accuracy rises monotonically across all four audited checkpoints (37.5%$\rightarrow$38.6%$\rightarrow$41.7%$\rightarrow$42.6%$\rightarrow$44.2% at steps 0/20/80/160/200) with parse-failure falling alongside it (2.2%$\rightarrow$0.7% at step 160, ticking back up slightly to 1.1% at step 200), the same benign pattern as AI2D's parser-failure reduction – across all three vision-language benchmarks, the large majority of the reported gain reflects the model genuinely answering more problems correctly. On MathVista, our exhaustive audit (not sampled) surfaced one narrow edge case worth flagging for transparency rather than a broad problem: the 1000-problem set contains 49 "age gap between two people in a photo" questions, and the steered model answers *exactly* 0 on all 49 regardless of the true gap, correct only on the single case where gold happens to be 0 (reached via a hallucinated "they are the same person" claim we independently falsified against the source photo); baseline is also 0/49 correct on this template, mostly via honest "cannot be determined" refusals. On this single audited run (net +6.2pp; three-seed mean +3.80 in Table 1), the template's net contribution is +1 problem out of 62 net-improved problems – too small to meaningfully drive the headline number, but a useful illustration that majority-vote pseudo-labels can occasionally converge on a fixed answer rather than genuine reasoning on a narrow template. It does not extend to any of the other 951 MathVista problems or to LogicVista.

**MMAU: the phoneme-counting confound.** The question template "count the number of words that contain at least one [un]stressed phoneme" appears 153/1000 times (15.3%) in the full set. Checked exhaustively (Table 7):

Roughly two-thirds (23/35) of the entire reported MMAU improvement traces to this one 15%-of-dataset template, on which neither model's stated reasoning logically derives its own stated answer:

| ID | Question (abridged) | Baseline → Steered | Gold | Verdict |
|---|---|---|---|---|
| **MathVista** | | | | |
| *Improved (baseline wrong → steered correct)* | | | | |
| 320 | Right-triangle midpoints; find $DE$ ($AB$=13, $AC$=5, $\angle C$=90°) | $DE$=$AB/2$=6.5 (wrong theorem) → $BC$=$\sqrt{13^2-5^2}$=12, $DE$=6 | 6 | Fixed; correct midsegment theorem |
| *Error (steered still wrong)* | | | | |
| 210 | Age gap between two people in a photo | wrong numeric guess → answers "0" | nonzero | One of 49 age-gap questions; see mode-collapse analysis below |
| **LogicVista** | | | | |
| *Improved (baseline wrong → steered correct)* | | | | |
| v1_203 | Compute a 2023 value after a 26% production increase from 2022 (MCQ) | arithmetic left incomplete mid-calculation, D (wrong) → finished the chain (20K × 0.26=5.2K, 20K+5.2K=25.2K), C (correct) | C | Fixed; completed the arithmetic chain baseline abandoned |
| *Hurt (baseline correct → steered wrong)* | | | | |
| v1_0 | Which choice completes a row/column shape pattern (MCQ) | correctly tracked the row/column pattern → C (correct) → re-opened the analysis, fixated on irrelevant dot positions → D (wrong) | C | Over-correction: doubted its own already-correct reasoning |
| **MMAU** | | | | |
| *Improved (baseline wrong → steered correct)* | | | | |
| 158 | Count words with ≥1 (un)stressed phoneme (153/1000 items are this template) | repetition-loop failure, empty prediction → "five" from 2 cited words | five | Confound: template alone drives ∼66% of net gain |
| *Error (steered still wrong, incl. regressions)* | | | | |
| 482 | Timestamp for a specific named chord | wrong answer, specific wording → near-identical wrong wording | – | Persistent error; suggests convergence on a shared wrong pattern |

Table 6: MathVista, LogicVista, and MMAU: 1 improved + 1 error/regression case each (MathVista rows drawn from 20 sampled cases; audio itself not directly verifiable for MMAU, audited for textual self-consistency and grading-code correctness). LogicVista rows are drawn from 9 manually-reviewed cases (7 of the 77 helped cases, 2 of the 47 hurt cases); the remaining reviewed helped cases span shape/pattern discrimination, sequence completion, spatial reasoning, and simple physical intuition, and the other reviewed hurt case similarly shows symbol misidentification breaking otherwise-correct logic. Images cross-checked against stated reasoning where applicable.

| | $n$ | Baseline | Steered | Δ |
|---|---|---|---|---|
| Phoneme subset | 153 | 40.5% | 55.6% | +15.1pp |
| Everything else | 847 | 60.4% | 61.9% | +1.4pp |
| **Full set** | **1000** | **57.4%** | **60.9%** | **+3.5pp** |

Table 7: MMAU-mini, phoneme-counting subset vs. the rest of the set.

e.g. one response lists three word-mentions ("director," "he's," "he's" again) as containing the target phoneme, then answers "five." Precise phoneme-stress counting from raw audio is not a capability either model plausibly has; the pattern resembles the steered model learning, via majority-vote RL, to avoid baseline's "count every word" heuristic in favor of smaller numbers that correlate better with this dataset's answers, without evidence of genuine phonetic analysis. An answer-letter distribution check rules out naive majority-class exploitation: gold answers skew toward `A` (37.7%), but predicted-`A` rate *falls* after steering (34.6%→31.1%) while predicted-`B` rises (23.2%→30.7%). Regressions occur on 79/1000 MMAU problems vs. 56/1000 on MathVista; several show near-identical wording between baseline and steered outputs (id 482 above), suggesting convergence on a shared wrong pattern we did not isolate to a single template.

**How the MMAU headline number should be read.** We want to be explicit about what this does and does not leave standing, since the main paper reports MMAU as one of six improved benchmarks. The +3.5pp headline gain on MMAU-mini is not uniformly distributed across the benchmark: it is concentrated on a single template covering 15.3% of the set, and once that template is excluded the remaining 847 problems improve by only +1.4pp. A gain of that size is within the seed-to-seed variation we observe elsewhere in this paper, so we do not claim a demonstrated improvement on the non-template portion of MMAU. The honest reading is that our audio result rests substantially on one question template whose gain we cannot attribute to genuine phonetic reasoning, and that MMAU is the weakest of our six benchmark results for this reason. We report

it because excluding a benchmark after seeing its audit would be a worse practice than reporting it with this caveat attached, and because the effect is diagnosable only through exactly this kind of per-template audit, which we would encourage for label-free test-time adaptation generally, since the reward provides no external signal that would flag such a shortcut during training. The MathVista confound examined above is an order of magnitude smaller (one net problem out of 62) and does not carry the same weight.

## D Labeled vs. Label-Free Vector Comparison: Full Detail

This appendix reports a full comparison between the learned label-free bias vector and a labeled bias-steering vector, examining whether label-free training produces a degenerate intervention or a task-specific change in model behavior.

Both compared vectors are all-layer `down_proj.bias` vectors on Qwen2.5-7B; the label-free vector is trained with GRPO on MATH-500 itself, while the labeled comparison vector is Sinii (Sinii et al., 2025)-reproduced, trained with RLOO on 40K labeled DeepScaleR problems and evaluated here out-of-distribution, so the reward source, optimizer, and training data all differ, not only the label source. Since the two checkpoints' originally-reported numbers used different context lengths, we re-evaluated both under one matched harness (same prompts, greedy decoding, context budget) on the same 500 MATH-500 problems: baseline 46.8% (matching Sinii-repro's own reported step-0 exactly, confirming harness parity), label-free 72.0%, labeled 75.6%. We additionally verified the baseline-vs-label-free delta's significance without relying on multi-seed training: a paired bootstrap (10,000 resamples) over this same 500-problem eval gives a 95% CI of $[+20.6, +29.8]$pp for the accuracy delta, and McNemar's exact test on the 166 discordant predictions gives $p < 10^{-20}$.

**Relation to the label-source ablation in the main paper.** This appendix and §6.2 answer two different questions, and their labeled-vs-label-free comparisons therefore point in different directions; we state the distinction here so that neither result is read as contradicting the other. The main-paper ablation (Table 2) is a controlled comparison in which *only* the reward source changes: both arms are our own runs, on the same MATH-500 problems, with the same optimizer, rollout count, learning rate, and step budget. Under that control, removing ground-truth labels costs nothing at this parameter budget, and the pseudo-label arm is 1.2–2.2 points ahead. The comparison in this appendix is not controlled in that sense: the labeled vector is a reproduction of a different method, trained with a different optimizer (RLOO) on a different and much larger dataset (40K DeepScaleR problems), and evaluated here out-of-distribution on MATH-500. Under those conditions it reaches 75.6% against the label-free vector's 72.0%. The two findings are consistent once the confound is accounted for: a label-free reward is not what costs accuracy relative to a labeled one at a matched recipe, but a labeled vector trained on $80\times$ more problems with a different optimizer can still be stronger on this evaluation. We report both rather than only the favourable one, and we do not claim that label-free bias-only TTRL outperforms labeled bias steering in general; only that, holding data and optimizer fixed, the label source is not the binding constraint.

A per-sample confusion matrix shows the two vectors agree on *which* problems they fix far more than chance would predict: 129/500 fixed by both, 26 labeled-only, 17 label-free-only, 4 regress under both, a Jaccard overlap of $129/(129{+}26{+}17){=}0.75$. Spot-checking the disagreements shows ordinary arithmetic slips on the losing side, not a different problem-solving strategy.

Cosine similarity between the two 28-layer, 3584-dimensional vectors shows a layer-wise trend: $+0.22$ to $+0.33$ in the earliest layers (0–19% depth, far from the $\approx 0$ expected for two random vectors, though the overall cosine of $+0.172$ is itself modest), falling to $+0.01$ and finally $-0.035$ by the final layer, evidence the two vectors converge toward similar computation early and grow checkpoint-specific late; Appendix E looks at the same divergence through a complementary logit-lens.

## E Logit-Lens Comparison: Full Detail

This appendix reports the full per-depth logit-lens token comparison against labeled bias steering, complementing the vector-comparison detail in Appendix D with additional model/task comparisons.

The full semantic progression on Qwen2.5-7B (label-free, MATH-500) shows layers 0–∼61% depth are semantic noise; "solve"/"solving" vocabulary emerges at ∼68–71% depth; `\boxed{}` becomes the top cosine-similarity token at ∼86–89% depth; and causal connectors ("Therefore", "Thus", "First") dominate the final layers. This task- and teacher-specific character is confirmed across three further comparisons, each varying only model or task: our labeled Sinii-reproduction run shows a distinct late-layer vocabulary, detailed below (Table 8); the Qwen2.5-Math-7B variant shows a different step-prefix cluster; and the COT-format MathVista and MMAU vectors (main paper §6.6) show modality-specific late-layer clusters rather than math-specific vocabulary.

| Depth | Label-free top tokens | Labeled (Sinii-repro) top tokens |
|---|---|---|
| 74% | `solve`, `solving` | noisy |
| 81% | `Solution`, `solve` | `simpl`, `simplify` |
| 89% | `solution`, `boxed` | noisy |
| 96% | `solve`, `boxed`, `Therefore/Thus` | noisy (non-English) |
| 100% (final) | noisy/formatting | `To`, `\`, `of` |

Table 8: Representative logit-lens tokens, label-free vs. labeled Sinii-reproduced (Sinii et al., 2025) (both Qwen2.5-7B, all-layer `down_proj.bias`), at matched depths.

The two vectors' tokens are disjoint at every depth in Table 8, consistent with the near-zero/negative cosine similarity at these depths (Appendix D). Label-free's late-layer tokens cluster around `Solution`/`solve`, `\boxed{}`, and step-connectives; labeled's cluster around `simplify`-family tokens at 81% depth and a step-introduction token plus a raw LaTeX-continuation marker at the final layer, evidence of different lexical biases despite similar reasoning trajectories; we do not take this as proof of what each vector computes.

**Format changes what the vector encodes, not just how well it performs.** The boxed-format MathVista vector (delayed-cosine, the checkpoint behind the main paper's MathVista headline of 65.40% in Table 1) gives a final-layer cluster qualitatively different from every other checkpoint: its top token is the digit `0` (cosine 0.24), alongside other digits, MCQ letters, and a backslash, literally the `\boxed{}` answer alphabet, rather than the abstract connectives or negation tokens seen in the text MATH-500 and COT-format MathVista checkpoints.

# F Additional Theoretical Validation Details

This appendix expands the main paper's theoretical grounding (Method §4.2) with full proof sketches, kept here for transparency and space rather than in the main text.

## F.1 Full Statements and Proof Sketches

**Proposition 1 setup.** For an unlabeled input $x$, let the model induce an answer distribution $p(a \mid x)$ over $K$ possible answers, with $a^\star$ the correct answer. Sample $G$ i.i.d. rollouts $A_1, \ldots, A_G \sim p(\cdot \mid x)$ and let $N_a = \sum_{g=1}^{G} \mathbf{1}[A_g = a]$; the majority-vote pseudo-label is $\hat{a}_G = \arg\max_a N_a$ (ties count as failures, a conservative choice). Define the answer margin $\Delta(x) = p(a^\star \mid x) - \max_{a \neq a^\star} p(a \mid x)$ and assume $\Delta(x) > 0$: the correct answer is the model's unique mode.

**Proposition 2** (Pseudo-Label Reliability)**.** *If* $\Delta(x) > 0$*,*

$$\Pr[\hat{a}_G \neq a^\star] \;\leq\; (K-1)\exp\left(-\frac{G\Delta(x)^2}{2}\right).$$

*Pseudo-label failure under majority voting decays exponentially in $G$ for positive-margin problems.*

*Proof.*

- For any $a \neq a^\star$, let $Z_g = \mathbf{1}[A_g = a^\star] - \mathbf{1}[A_g = a] \in [-1, 1]$.
- Then $\mathbb{E}[Z_g] = p(a^\star \mid x) - p(a \mid x) \geq \Delta(x) > 0$ and $\sum_g Z_g = N_{a^\star} - N_a$.
- Answer $a$ can only tie or beat $a^\star$ if $\frac{1}{G}\sum_g Z_g \leq 0$.
- Hoeffding's inequality bounds this event by $\exp(-G\Delta(x)^2/2)$.

- A union bound over the $K-1$ incorrect answers gives the proposition.

□

**Theorem 1 setup.** Let $J(w)$ be the RL objective (the same advantage-weighted log-probability objective formalized in Algorithm 1) and $g = \nabla J(w)$ the full gradient. Restricting adaptation to a bias subspace $\mathcal{S}$ with orthogonal projection $P_{\mathcal{S}}$, only the component $P_{\mathcal{S}}g$ is reachable; define the accessible gradient energy $E_{\mathcal{S}} = \|P_{\mathcal{S}}g\|^2$ and the restricted update $w^+ = w + \eta P_{\mathcal{S}}g$.

**Theorem 2** (Restricted-Subspace Trainability)**.** *If $J$ is locally $L$-smooth,*

$$J(w^+) - J(w) \;\geq\; \eta\Big(1 - \tfrac{L\eta}{2}\Big)\|P_{\mathcal{S}}g\|^2,$$

*and for $0 < \eta \leq 1/L$, $J(w^+) - J(w) \geq \frac{\eta}{2}\|P_{\mathcal{S}}g\|^2$ (the bound reported in the main paper).*

*Proof.*

- $L$-smoothness gives $J(w+d) \geq J(w) + \langle \nabla J(w), d\rangle - \frac{L}{2}\|d\|^2$.
- Set $d = \eta P_{\mathcal{S}}g$ and use $\langle g, P_{\mathcal{S}}g\rangle = \|P_{\mathcal{S}}g\|^2$ (a projection is idempotent and self-adjoint) to get

  $$J(w^+) - J(w) \geq \eta\|P_{\mathcal{S}}g\|^2 - \tfrac{L\eta^2}{2}\|P_{\mathcal{S}}g\|^2,$$

  which is the stated bound.
- The $\eta \leq 1/L$ case follows by bounding its coefficient below by $1/2$.
- Writing $\rho_{\mathcal{S}} = \|P_{\mathcal{S}}g\|^2/\|g\|^2$ for the accessible-gradient fraction, $J(w^+) - J(w) \geq \eta(1 - \frac{L\eta}{2})\rho_{\mathcal{S}}\|g\|^2$: local improvement scales with $E_{\mathcal{S}}$, not with $\dim(\mathcal{S})$.

□