TTSHIFT: LEARNING TO ADAPT AT TEST TIME WITH SHARED SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlabeled test-time learning adapts large language models using objectives computed from target inputs, but these objectives do not directly supervise answer correctness. We show that this mismatch can affect reasoning behavior: input next-token cross-entropy can continue to decrease even as target accuracy and off-target behavior deteriorate. This raises a central question: how much update freedom is necessary to turn an indirect test-time objective into useful behavioral change? We introduce TTSHIFT (**T**est-**T**ime **Shift**), which freezes pretrained linear operators and learns dataset-shared additive shifts at Q/V projection outputs from unlabeled target inputs. With a single fixed learning rate, TTSHIFT increases equal-task mean accuracy across seven reasoning tasks from 45.11% to 48.49% on Qwen3-8B-Base and from 55.54% to 56.48% on Qwen3.5-4B-Base, improvements of 3.38 and 0.94 percentage points. In a matched ten-rate Q/V sweep on Qwen3-8B-Base, TTSHIFT achieves positive gains on 68.57% of task-rate configurations, compared with 31.43% for rank-8 LoRA. Thus, useful adaptation persists after removing learned input-dependent mapping corrections and remains available across a broader range of update strengths. Training-dynamics analysis further shows that the shared update direction becomes organized under partial cross-example gradient agreement before its full magnitude accumulates. These results show that effective test-time adaptation can emerge from a highly restricted update space, making update freedom a concrete design dimension for selective plasticity under indirect supervision. Anonymous code for reproducing the main experiments is available at `https://anonymous.4open.science/r/review-code-FDE3/`.

1 INTRODUCTION

Test-time learning adapts large language models (LLMs) to unlabeled deployment inputs, but its objectives provide only indirect supervision of answer quality. Next-token cross-entropy (CE), for example, can decrease even as reasoning accuracy or existing capabilities deteriorate. Although TLM constrains test-time updates with LoRA (Hu et al., 2025a), is parameter efficiency sufficient, or does the *form* of the update matter? In a ten-learning-rate Q/V LoRA sweep on MATH500, GPQA Diamond, and LogiQA, stronger CE optimization can coincide with worse target accuracy and preservation. On GPQA, increasing the learning rate from 10^{-5} to 3×10^{-5} changes the target gain from $+6.06$ pp to -11.11 pp while off-target decline D rises from 10.41 pp to 45.23 pp. Thus, LoRA alone does not ensure that better surrogate optimization yields better behavioral adaptation. This motivates *selective plasticity*: improving target behavior while limiting unnecessary drift. Yet LoRA, despite its parameter efficiency, still learns an input-dependent correction (Hu et al., 2022),

$$XW \to XW + XBA.$$

We ask whether this flexibility is necessary and instead consider the more restrictive update

$$XW \to XW + \beta,$$

where W remains frozen and β is shared across target examples and token positions. Despite removing the learned input-dependent correction, this shared shift retains useful adaptation. On GPQA, LoRA and the shared shift both reach $+3.03$ pp target gain, with off-target costs of 1.876 pp and 0.257 pp, respectively; at approximately 2 pp cost, their gains are $+3.03$ pp and $+7.58$ pp. LogiQA

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.

shows the same qualitative advantage. These results suggest that restricting the *form* of the update can change how an indirect test-time objective translates into behavior.

Building on this observation, we introduce TTSHIFT (**T**est-**T**ime **Shift**), which freezes pretrained linear operators and learns lightweight, dataset-shared additive shifts from unlabeled inputs. We use Q/V projection-output shifts as the main configuration. At a fixed learning rate of 8×10^{-5}, TTSHIFT raises equal-task mean accuracy across seven reasoning tasks from 45.11% to 48.49% on Qwen3-8B-Base ($+3.38$ pp) and from 55.54% to 56.48% on Qwen3.5-4B-Base ($+0.94$ pp). We further find that the shared update direction emerges under partial per-example gradient alignment and stabilizes before its magnitude fully accumulates.

We make three contributions. First, we show that test-time CE optimization can diverge from both target accuracy and preservation, motivating selective plasticity for unlabeled adaptation. Second, controlled Q/V comparisons show that useful adaptation persists after removing LoRA's learned input-dependent correction, with shared shifts reaching favorable gain–preservation regimes. Third, we introduce TTSHIFT, demonstrate improvements across two Qwen base models and seven reasoning tasks, and characterize how its shared update direction emerges during optimization.

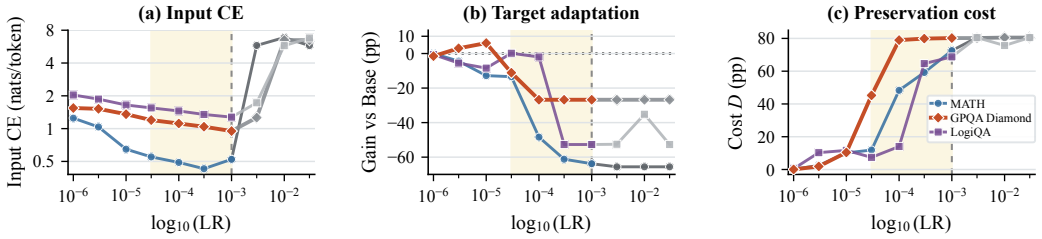

Figure 1: **Surrogate–behavior divergence under test-time LoRA.** One-epoch endpoints from a ten-rate Q/V LoRA sweep. **(a)** Target CE. **(b)** Target accuracy change from Base. **(c)** Off-target accuracy decline D. The yellow region shows CE improvement despite behavioral degradation; gray marks larger-rate settings where CE becomes unstable.

2 Related Work

Test-Time Adaptation. Test-time adaptation (TTA) uses unlabeled deployment inputs to improve pretrained models under distribution shift. TTT introduces an auxiliary self-supervised task, while Tent adapts normalization parameters through entropy minimization (Sun et al., 2020; Wang et al., 2021). Another architectural approach treats the recurrent hidden state as a model and updates it through self-supervised learning while processing a test sequence (Sun et al., 2025). EATA filters unreliable or redundant samples and regularizes against forgetting; COME models uncertainty to make entropy updates more conservative (Niu et al., 2022; Zhang et al., 2024). For LLMs, TLM updates LoRA from input likelihood, whereas SLOT optimizes a per-question vector at the final hidden representation using prompt CE (Hu et al., 2025a;b). Other approaches derive supervision from majority-vote generations, semantic entropy, or stability-aware confidence (Zuo et al., 2025; Zhang et al., 2025; Gu et al., 2026). We use ordinary next-token CE and compare update classes under this indirect signal when answer labels are unavailable.

Representation Adaptation. Parameter-efficient methods differ in what they allow a pretrained model to change. LoRA learns low-rank weight corrections, BitFit updates biases, $(\mathrm{IA})^3$ rescales activations, and SSF, RED, and ReFT apply structured representation changes (Hu et al., 2022; Ben Zaken et al., 2022; Liu et al., 2022; Lian et al., 2022; Wu et al., 2024a;b). Activation steering adds directions to frozen-model representations: CAA and instruction-based steering derive them from behavioral contrasts or instructions, while DISCO intervenes in query/value spaces and AUSteer selects finer-grained activation units (Rimsky et al., 2024; Stolfo et al., 2024; Torop et al., 2025; Feng et al., 2026). Additive attention-space intervention is established; TTSHIFT learns dataset-shared, token-uniform Q/V translations directly from unlabeled target inputs with input CE. Unlike SLOT's per-question final-hidden vector (Hu et al., 2025b) or TASCO's task-level input prefix (Gu

et al., 2026), TTSHIFT adapts internal attention representations while keeping the pretrained projection matrices fixed. We treat this translation as a restricted *adaptation space* and test whether it provides sufficient plasticity under indirect test-time supervision.

3 WHY SHARED SHIFTS FOR TEST-TIME ADAPTATION?

Unlabeled test-time adaptation optimizes an observable surrogate, while deployment requires correct answers. We first examine how these objectives relate as update strength varies, then use controlled probes to study which behaviors change.

3.1 THE TEST-TIME SURROGATE IS AN IMPERFECT PROXY FOR LLM PERFORMANCE

In test-time learning for LLMs, target inputs are available at deployment but their gold answers are not. TLM (Hu et al., 2025a) uses next-token likelihood from unlabeled inputs to adapt lightweight LoRA parameters. This creates a mismatch: input cross-entropy (CE) is directly optimizable, whereas answer correctness is unavailable to the optimizer. We examine how closely this surrogate tracks reasoning behavior. Following the LoRA setting, we apply rank-8 Q/V LoRA to Qwen3-8B-Base (Qwen Team, 2025) and sweep ten learning rates from 10^{-6} to 3×10^{-2} for one epoch, keeping all other settings fixed. Adaptation and target evaluation use MATH500 (Hendrycks et al., 2021; Lightman et al., 2023) (500), GPQA Diamond (Rein et al., 2023) (198), and LogiQA (Liu et al., 2020) (651). We additionally evaluate GSM8K (Cobbe et al., 2021), OpenBookQA (Mihaylov et al., 2018), and LSAT-LR (Zhong et al., 2023) as off-target probes that never enter adaptation. Their degradation is summarized by D, the mean positive accuracy decline across the three tasks, preventing improvements on one probe from offsetting degradation on another.

Surrogate improvement can separate from reasoning behavior. In the yellow region of Figure 1, reasoning behavior deteriorates before surrogate optimization becomes unstable. On GPQA, a learning rate of 10^{-5} reduces input CE to 1.357 nats/token while improving target accuracy by $+6.06$ pp over Base. Increasing the rate to 3×10^{-5} lowers CE further to 1.198, yet target accuracy reverses to -11.11 pp, while off-target degradation rises from $D = 10.41$ pp to 45.23 pp. We refer to this region, where the observable surrogate continues to improve while reasoning behavior deteriorates, as *surrogate-behavior divergence*. This is distinct from the gray region at still larger learning rates, where the adaptation becomes much more aggressive and the surrogate optimization itself eventually becomes unstable.

The divergence is also visible at the question level. The aggregate accuracy change combines questions repaired and questions damaged. On GPQA, the 10^{-5} update repairs 35 Base-wrong answers while damaging 23 Base-correct ones. At 3×10^{-5}, only 19 answers are repaired while 41 previously correct answers become wrong. The adaptation therefore moves from a repair-dominated to a damage-dominated regime even as input CE continues to decrease. The response depends on the task: MATH500 shows no positive-gain operating point in this sweep, while LogiQA exhibits only a small positive region.

Observation 1. *The observable test-time surrogate and downstream reasoning behavior can diverge. Continued surrogate improvement can coincide with lower target accuracy, a transition from repair-dominated to damage-dominated answer changes, and greater degradation on tasks outside the adaptation objective.*

3.2 PROBING UPDATE FREEDOM AT TEST TIME

Given the surrogate–behavior divergence in Section 3.1, we ask how much update freedom the model needs under an indirect test-time objective. TLM (Hu et al., 2025a) already restricts test-time updates by using LoRA rather than full-parameter optimization. We ask whether useful adaptation requires LoRA's learned input-dependent correction, or whether a simpler shared translation provides sufficient flexibility.

From an input-dependent correction to a shared translation. Consider an input representation X_l at a query or value projection $W_{l,m}$, where $m \in \{q, v\}$. LoRA freezes the pretrained projection

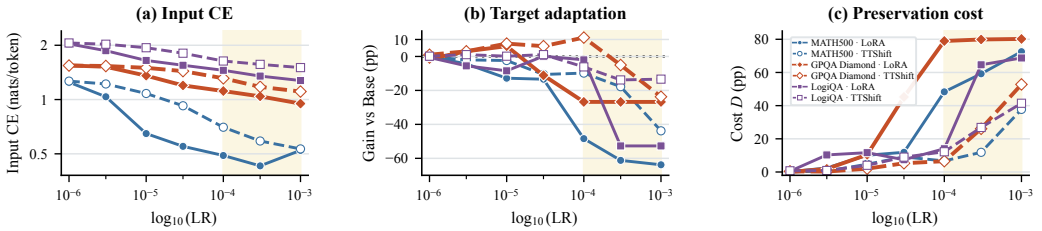

Figure 2: **Test-time adaptation across learning rates.** Q/V LoRA and the shared-shift probe are evaluated on MATH500, GPQA Diamond, and LogiQA under the same adaptation protocol. **(a)** Final target input CE after adaptation (logarithmic y-axis). **(b)** Target accuracy gain relative to Base. **(c)** Mean positive accuracy decline D across the three off-target probe tasks. As adaptation becomes stronger, LoRA often achieves larger CE reductions, while the shared shift retains a broader region of useful target behavior with lower measured preservation cost.

and learns a low-rank correction,

$$Y_{l,m}^{\mathrm{LoRA}} = X_l W_{l,m} + X_l B_{l,m} A_{l,m},$$
$$Y_{l,m}^{\mathrm{shift}} = X_l W_{l,m} + \mathbf{1}\beta_{l,m}^{\top}. \tag{1}$$

The LoRA correction $X_l B_{l,m} A_{l,m}$ depends on the current representation, allowing the learned modification to vary with the input. We remove this input dependence and retain only a translation $\beta_{l,m}$ shared across examples and token positions, while keeping the pretrained projection frozen. This restriction lets us probe how much adaptation freedom is needed.

We compare rank-8 Q/V LoRA and the shared-shift probe at the same Q/V projection outputs, using the same unlabeled next-token CE objective, target inputs, optimizer, data order, seed, one-epoch budget, and greedy evaluation. Both are evaluated over the same learning-rate sweep on MATH500, GPQA Diamond, and LogiQA, with the three off-target tasks from Section 3.1 used to measure preservation cost D.

A restricted update still supports useful adaptation. Figure 2 shows that removing the input-dependent correction does not prevent the model from responding to the test-time objective. The shared shift continues to reduce input CE as adaptation strength increases and reaches positive target gains on GPQA and LogiQA. On GPQA, its best measured operating points exceed the gains obtained by LoRA in the same sweep. Neither update form produces a positive-gain point on MATH500, so the extent of useful adaptation still depends on the task. Across these tasks, useful adaptation can persist despite a substantial restriction on the form of the learned update.

Update freedom shapes how surrogate pressure translates into behavior. The update forms differ more as adaptation becomes stronger. LoRA often continues to reduce input CE more aggressively, while its target accuracy and off-target preservation deteriorate rapidly. The shared shift also responds to stronger optimization, but its behavioral degradation is more moderate over the measured sweep. This produces a broader range of operating points in which surrogate improvement remains behaviorally useful.

The measured configurations quantify this difference. On GPQA, both update forms attain a $+3.03$ pp target gain, yet the corresponding preservation cost is 0.257 pp for the shared shift and 1.876 pp for LoRA. At a similar preservation cost of approximately 2 pp, the shared shift instead reaches $+7.58$ pp, compared with $+3.03$ pp for LoRA. LogiQA shows the same direction at a smaller gain: around $+0.15$ pp, the measured preservation costs are 0.725 pp and 7.394 pp for the shared shift and LoRA, respectively. These operating points suggest that useful adaptation and a more moderate collateral response to stronger surrogate optimization can coexist under a restricted update.

The probe suggests that a dataset-shared translation retains substantial adaptation capacity despite its simple form. Useful gains persist with less update freedom, and behavior can remain favorable over a wider part of the measured adaptation range. Update freedom is therefore a design choice; a more expressive test-time correction is not always necessary.

Observation 2. *Useful test-time adaptation can persist in a highly restricted update space. Across the measured sweep, shared shifts retain positive adaptation while exhibiting a broader region in which stronger surrogate optimization remains behaviorally useful. This suggests that update freedom influences how test-time surrogate pressure is expressed in model behavior.*

3.3 From Shared Translation to TTShift

Design principle. The controlled probe in Section 3.2 shows that useful test-time adaptation can persist even after removing the learned input-dependent correction. Our design principle is: *freeze the pretrained linear operators and learn only a shared additive translation*. For a selected projection site s, let $f_s(\cdot; W_s)$ denote the pretrained projection with fixed weight matrix W_s and $z_{s,t}$ its input at token position t. TTShift modifies its output as

$$h_{s,t} = f_s(z_{s,t}; W_s) \quad \longrightarrow \quad \widetilde{h}_{s,t} = h_{s,t} + \beta_s, \qquad W_s \text{ frozen}, \quad \beta_s^{(0)} = 0. \tag{2}$$

Each $\beta_s \in \mathbb{R}^{d_s}$ is shared across token positions and examples in the target dataset, and is reinitialized for a new target dataset. The learned change is a simple dataset-level translation. The frozen network can transform it into input-dependent downstream behavior.

Where should the shared translation be applied? We also need to choose where to apply the translation. We compare translations at Q/V projection outputs, the attention-branch output after W_O, the block output after the MLP residual, and all Q/K/V/O projection outputs. All placements use the same frozen Qwen3-8B-Base, unlabeled input-CE objective, seven learning rates from 10^{-6} to 10^{-3}, and the same off-target preservation probes. Figure 3 summarizes the measured GPQA operating points under increasing preservation-cost budgets.

The placements differ in the low-cost operating region. On GPQA, Q/V shifts reach $+3.03$ pp at $D \leq 1$ pp and $+7.58$ pp at $D \leq 3$ pp, compared with $+1.52$ and $+7.07$ pp for attention-output shifts, and still smaller gains for the other placements under the same budgets. Q/V also reaches $+11.11$ pp at a measured cost of 6.53 pp, whereas the attention-output placement requires 9.04 pp to reach the same gain. The pattern extends to LogiQA: under $D \leq 1$ pp, Q/V reaches $+1.08$ pp, compared with $+0.46$ pp for attention-output, $+0.15$ pp for block-output, and no positive measured point for Q/K/V/O. MATH500 yields no positive-gain point for any of the four placements in this sweep, so the benefit still depends on the task.

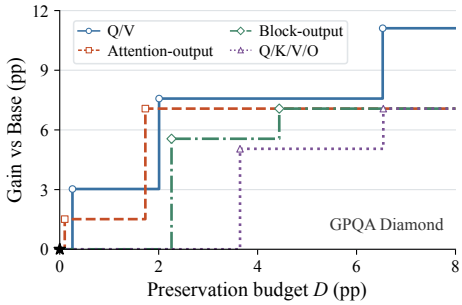

Figure 3: **GPQA gain under preservation-cost budgets.** Each curve summarizes the best measured target gain attainable within a given preservation-cost budget over the common seven-rate sweep. The comparison is descriptive over observed configurations.

These results suggest that intervention location affects both the attainable target improvement and its preservation cost. In the measured low-cost region, Q/V translations provide consistently favorable operating points across GPQA and LogiQA. We therefore use Q/V projection outputs as the primary TTShift configuration in the remainder of the paper.

TTShift learning rule. For each layer l, TTShift learns a query shift $\beta_{l,q}$ and a value shift $\beta_{l,v}$ while keeping all pretrained parameters frozen. The shifts are added directly to the corresponding projection outputs and are shared across examples and token positions. Given an unlabeled target dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, where each x_i contains the observed question, answer choices when present, and a fixed answer-format instruction, we optimize next-token cross-entropy:

$$\min_{\beta} \; \mathcal{L}(\mathcal{D}; \beta) = \frac{1}{N} \sum_{i=1}^{N} \left[-\frac{1}{T_i - 1} \sum_{t=2}^{T_i} \log p_{\theta,\beta}(x_{i,t} \mid x_{i,<t}) \right]. \tag{3}$$

Gold answers, pseudo-labels, and generated solutions do not enter the objective. We optimize the shared shifts for one shuffled epoch and retain the final offsets for evaluation. The main experiments use Adam with a fixed learning rate of 8×10^{-5}.

For Qwen3-8B-Base, learning Q/V shifts across all 36 layers introduces only 184,320 trainable parameters. Rank-8 LoRA at the same Q/V projections uses 3,833,856 parameters, so TTSHIFT uses approximately 4.8% as many trainable parameters. The restriction also concerns the form of the update: TTSHIFT keeps the pretrained projection matrices fixed and learns only shared additive translations, without learned input-dependent correction terms.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Setup. We evaluate Qwen3-8B-Base (Qwen Team, 2025) and Qwen3.5-4B-Base (Qwen Team, 2026) on seven reasoning tasks: Minerva Math (Lewkowycz et al., 2022), LSAT-AR from AGIEval (Zhong et al., 2023), MuSR Object Placements (Sprague et al., 2024), ARC-Challenge (Clark et al., 2018), SVAMP (Patel et al., 2021), AIME25 (Zhang & Math-AI Team, 2025), and the Philosophy subset of MMLU-Pro (Wang et al., 2024), totaling 2,759 questions. For each task, we use the same unlabeled target questions for adaptation and evaluation in a separate transductive setting. For TTSHIFT, all pretrained weights are frozen and dataset-shared additive shifts are learned at the Q/V projection outputs using next-token CE on the unlabeled target inputs. Each task is adapted for one epoch with Adam at a fixed learning rate of 8×10^{-5}. The Q/V LoRA baselines use rank 8, zero dropout, and a fixed learning rate of 10^{-6}; other baselines follow their respective recipes. We generate one greedy completion per question, count invalid or unparseable answers as incorrect, and report per-task accuracy, percentage-point change relative to each model's own Base performance, and the equal-weight macro mean across the seven tasks. Unless otherwise noted, results are single-seed point estimates. Appendix A gives dataset details and full training and baseline hyperparameters.

Table 1: **Main results across two models and seven reasoning tasks.** Greedy accuracy (%); small annotations denote percentage-point changes from Base.

Dataset	Base	LoRA	TLM	SLOT	EATA	TTSHIFT
Qwen3-8B-Base						
Minerva Math	22.43	22.79$_{\uparrow+0.37}$	24.26$_{\uparrow+1.84}$	23.16$_{\uparrow+0.74}$	22.43$_{\pm0.00}$	**28.31**$_{\uparrow+5.88}$
LSAT-AR	26.96	27.39$_{\uparrow+0.43}$	26.52$_{\downarrow-0.43}$	28.26$_{\uparrow+1.30}$	26.52$_{\downarrow-0.43}$	**30.87**$_{\uparrow+3.91}$
MuSR Object Placements	43.75	42.19$_{\downarrow-1.56}$	43.36$_{\downarrow-0.39}$	**47.27**$_{\uparrow+3.52}$	43.36$_{\downarrow-0.39}$	44.92$_{\uparrow+1.17}$
ARC-Challenge	89.16	**90.78**$_{\uparrow+1.62}$	89.25$_{\uparrow+0.09}$	88.65$_{\downarrow-0.51}$	89.59$_{\uparrow+0.43}$	90.27$_{\uparrow+1.11}$
SVAMP	91.33	91.00$_{\downarrow-0.33}$	92.00$_{\uparrow+0.67}$	91.67$_{\uparrow+0.33}$	**92.33**$_{\uparrow+1.00}$	92.00$_{\uparrow+0.67}$
AIME25	6.67	**10.00**$_{\uparrow+3.33}$	6.67$_{\pm0.00}$	3.33$_{\downarrow-3.33}$	6.67$_{\pm0.00}$	**10.00**$_{\uparrow+3.33}$
MMLU-Pro Philosophy	35.47	32.87$_{\downarrow-2.61}$	36.67$_{\uparrow+1.20}$	24.25$_{\downarrow-11.22}$	33.47$_{\downarrow-2.00}$	**43.09**$_{\uparrow+7.62}$
Macro mean	45.11	45.29$_{\uparrow+0.18}$	45.53$_{\uparrow+0.42}$	43.80$_{\downarrow-1.31}$	44.91$_{\downarrow-0.20}$	**48.49**$_{\uparrow+3.38}$
Qwen3.5-4B-Base						
Minerva Math	41.54	42.28$_{\uparrow+0.74}$	42.28$_{\uparrow+0.74}$	42.65$_{\uparrow+1.10}$	43.01$_{\uparrow+1.47}$	**43.38**$_{\uparrow+1.84}$
LSAT-AR	40.87	40.87$_{\pm0.00}$	38.26$_{\downarrow-2.61}$	40.00$_{\downarrow-0.87}$	40.43$_{\downarrow-0.43}$	**43.04**$_{\uparrow+2.17}$
MuSR Object Placements	42.97	42.58$_{\downarrow-0.39}$	42.19$_{\downarrow-0.78}$	44.14$_{\uparrow+1.17}$	44.92$_{\uparrow+1.95}$	**45.31**$_{\uparrow+2.34}$
ARC-Challenge	95.48	**95.82**$_{\uparrow+0.34}$	95.73$_{\uparrow+0.26}$	95.73$_{\uparrow+0.26}$	95.22$_{\downarrow-0.26}$	95.73$_{\uparrow+0.26}$
SVAMP	**91.33**	**91.33**$_{\pm0.00}$	90.67$_{\downarrow-0.67}$	89.33$_{\downarrow-2.00}$	90.67$_{\downarrow-0.67}$	**91.33**$_{\pm0.00}$
AIME25	16.67	16.67$_{\pm0.00}$	16.67$_{\pm0.00}$	20.00$_{\uparrow+3.33}$	**23.33**$_{\uparrow+6.67}$	16.67$_{\pm0.00}$
MMLU-Pro Philosophy	59.92	58.92$_{\downarrow-1.00}$	59.52$_{\downarrow-0.40}$	**61.32**$_{\uparrow+1.40}$	57.52$_{\downarrow-2.40}$	59.92$_{\pm0.00}$
Macro mean	55.54	55.49$_{\downarrow-0.05}$	55.04$_{\downarrow-0.50}$	56.17$_{\uparrow+0.63}$	56.44$_{\uparrow+0.90}$	**56.48**$_{\uparrow+0.94}$

4.2 TEST-TIME REASONING ADAPTATION RESULTS

Using the protocol in Section 4.1, we evaluate the final Q/V TTSHIFT configuration across two model scales and seven reasoning tasks. Table 1 reports the results together with each model's Base performance and the available test-time adaptation baselines.

Across tasks. On Qwen3-8B-Base, TTSHIFT improves all seven reasoning tasks over Base, increasing the equal-task macro mean from 45.11 to 48.49, a gain of +3.38 percentage points. Gains across the evaluation set include +5.88 pp on Minerva Math, +3.91 pp on LSAT-AR, +3.33 pp on AIME25, and +7.62 pp on MMLU-Pro Philosophy. TTSHIFT achieves the highest displayed accuracy on three tasks and ties for the highest on AIME25, while maintaining positive gains on the remaining tasks. The resulting macro mean of 48.49 is the highest among the displayed methods.

Across model scales. The Q/V-shift design principle gives consistent results on Qwen3.5-4B-Base. TTSHIFT improves four of the seven tasks and preserves Base performance on the remaining three, raising the macro mean from 55.54 to 56.48. It achieves the highest displayed accuracy on Minerva Math, LSAT-AR, and MuSR Object Placements. All seven tasks use a single fixed learning rate, without task-specific tuning. The gains therefore transfer across both task types and model scales.

Comparison with test-time adaptation baselines. TTSHIFT remains competitive with the test-time adaptation mechanisms in the table and achieves the highest displayed macro mean on both models using only dataset-shared Q/V offsets and frozen pretrained projection matrices. On Qwen3-8B-Base, the margin over Base reaches +3.38 pp, compared with +0.18 pp for LoRA and +0.42 pp for TLM. On Qwen3.5-4B-Base, TTSHIFT reaches a macro mean of 56.48, compared with 56.44 for EATA and 56.17 for SLOT. A simple dataset-shared representation translation can thus provide consistently useful reasoning adaptation across the evaluated tasks and model scales.

4.3 Stability Across Adaptation Strengths

We assess stability beyond the fixed operating point by varying adaptation strength. On Qwen3-8B-Base, we evaluate Q/V TTSHIFT and matched rank-eight Q/V LoRA across the same seven tasks and ten learning rates from 10^{-6} to 10^{-3}, yielding 70 task–learning-rate configurations per method. TTShift and LoRA use the same unlabeled input-CE objective, Q/V adaptation locations, optimizer, one-epoch training budget, data order, and evaluation protocol.

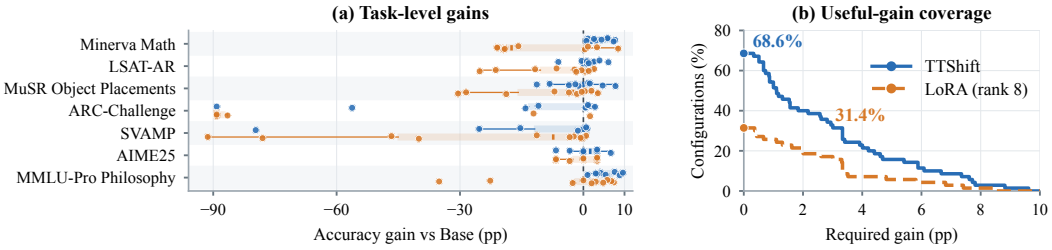

Figure 4: **Stability across adaptation strengths. Left:** accuracy change from Base across the measured grid. Each point is one task–learning-rate configuration. **Right:** fraction of the 70 configurations exceeding each gain threshold. At zero gain, TTSHIFT improves over Base in 48/70 configurations, versus 22/70 for LoRA.

Figure 4 compares the two update spaces across learning rates beyond the main setting. TTSHIFT improves over Base in 48/70 configurations (68.57%), compared with 22/70 (31.43%) for LoRA, so its positive operating region is substantially broader. The lower tail also differs: across all 70 configurations, the tenth-percentile gain is −10.96 pp for TTSHIFT versus −89.16 pp for LoRA. Thus, as the optimization scale moves away from favorable settings, the restricted translation parameterization remains beneficial more often and degrades much less severely when adaptation becomes harmful. We also examine item-level changes at the shared learning rate of 8×10^{-5}. We call a Base-incorrect example that becomes correct after adaptation a *repair*, and a Base-correct example that becomes incorrect a *damage*. Among examples answered correctly by Base, the equal-task average damage rate is 19.9% for TTSHIFT, compared with 55.1% for LoRA. This suggests that the broader operating region of TTSHIFT is accompanied by substantially less overwriting of predictions that the base model already gets right. These results show that a dataset-shared Q/V translation remains effective over a broad range of adaptation strengths while limiting destructive changes to

existing behavior. In the evaluated settings, a restricted update space can provide sufficient plasticity for useful test-time adaptation.

4.4 How Does a Dataset-Shared Shift Emerge?

All examples in a target dataset share the same Q/V translation vectors under TTSHIFT. We examine how this dataset-level direction emerges from heterogeneous example-level optimization signals. Across all seven tasks, we repeat the fixed Qwen3-8B-Base adaptation recipe and save checkpoints throughout the single adaptation epoch. At each checkpoint, we measure agreement among per-example gradients of the shared shifts using the normalized gradient consensus

$$C_2 = \frac{\|\sum_i g_i\|_2}{\sum_i \|g_i\|_2}, \tag{4}$$

together with leave-one-out cosine similarity, and track how the learned shift approaches its final direction and magnitude. Measurement definitions and the norm-robustness check are provided in Appendix B.

Figure 5 shows two consistent patterns. First, a shared direction can emerge without strongly aligned example gradients. At initialization, the median joint-QV consensus across tasks is only 0.379, while the aggregate leave-one-out cosine remains positive on all seven tasks. This indicates that heterogeneous gradients can still contain a common directional component that accumulates into a dataset-shared update. Second, this direction becomes recognizable before the update magnitude and behavioral effects stabilize. By 50% of adaptation, the median cosine to the final shift already reaches 0.916, while the shift has accumulated only 62.5% of its final norm. Repairs and damage continue to evolve thereafter, showing that a largely established direction can still produce different behavioral outcomes as the update grows.

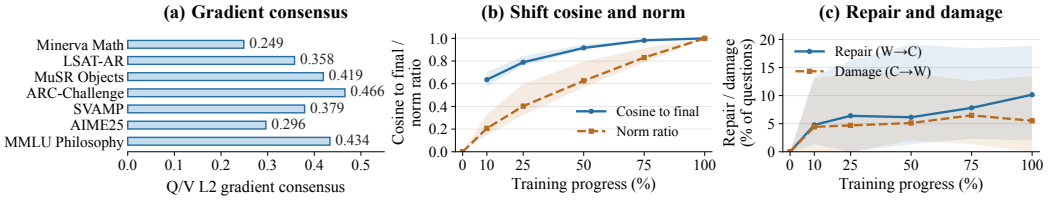

Figure 5: **Formation of a dataset-shared Q/V shift across seven tasks. (a)** Gradient consensus at initialization. **(b)** Cosine to the final Q/V shift and fraction of its final norm over adaptation. **(c)** Repairs and damage relative to Base. Lines show task medians and shading the seven-task min/max range. The shared direction becomes substantially aligned with its final orientation before its magnitude and behavioral effects stabilize.

The dataset-shared shift appears to accumulate a common component of heterogeneous example gradients, even without strong example-wise alignment. Its direction emerges relatively early, while its magnitude and repair–damage balance continue to evolve. These trajectories are descriptive and do not imply discrete optimization stages or a causal role for norm growth.

4.5 How Do Learned Q/V Shifts Alter Attention Computation?

We examine how simple dataset-shared translations change computation inside the frozen Transformer. To isolate the two components, we mask either the learned Q or V shifts at inference time while reusing the same jointly trained checkpoint. These Q-only and V-only variants are diagnostic interventions rather than separately optimized models. Both components affect predictions across the seven tasks, and their relative contributions depend on the task. We focus on how pretrained attention computation transforms each shared translation.

The learned Q shift changes attention routing without imposing a fixed semantic preference. We record attention logits after Q/K normalization and rotary position encoding, together with the resulting post-softmax probabilities. Across tasks, the Q shift consistently changes attention allocation, yet the signed effect varies across layers and prompt regions. For example, on MMLU-Pro

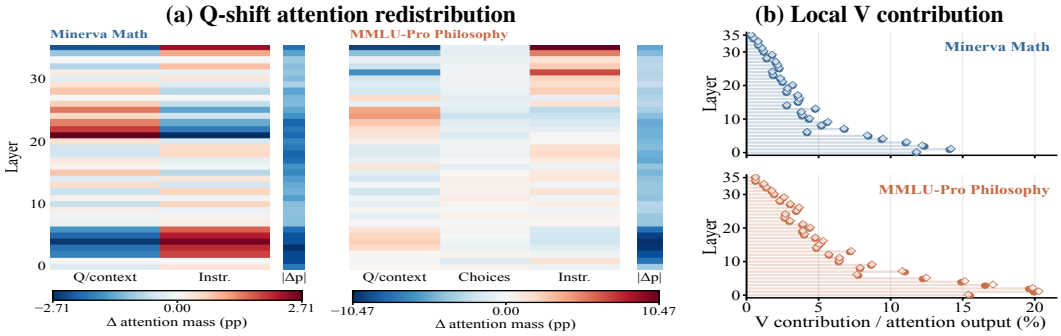

Figure 6: **How shared Q/V shifts are transformed by frozen attention.** **(a)** Q shifts produce layer- and region-dependent attention redistribution. **(b)** V shifts have a simple local post-W_O effect that becomes richer after propagation.

Philosophy, attention from the final query to the question/context, answer choices, and instruction each alternates between increases and decreases across the 36 layers; the same qualitative pattern appears on Minerva Math. These reversals show that the shared Q translation does not impose a fixed instruction to "attend more" to one region. Layer-specific representations and pretrained attention geometry repeatedly modulate its effect, producing routing changes that depend on the layer and task.

The measured forward pass shows a simple additive contribution from the shared V translation immediately after the output projection. The effect becomes more complex as it propagates. Once the modified attention output enters the residual stream, subsequent hidden states, queries, keys, and attention weights also change. The same dataset-shared V shift can therefore generate increasingly input-dependent effects as it propagates through the frozen network.

The frozen computation transforms a highly restricted update into expressive behavioral changes. TTSHIFT learns only dataset-shared translations at the Q/V projection outputs, whose effects depend on the pretrained computation through which they propagate. This simple adaptation space retains the expressive, input-dependent transformations of the frozen model.

5 CONCLUSION

We introduced TTSHIFT, which adapts reasoning behavior through dataset-shared Q/V translations learned from unlabeled inputs while freezing pretrained linear operators. Removing learned input-dependent corrections retains useful adaptation: at one fixed learning rate, TTSHIFT improves seven-task mean accuracy by 3.38 pp on Qwen3-8B-Base and 0.94 pp on Qwen3.5-4B-Base. Controlled probes show favorable gain–preservation operating points, while the matched ten-rate study finds a broader positive region and a milder lower tail than rank-8 Q/V LoRA. Training analysis shows that the shared direction forms under partial gradient agreement before its magnitude fully accumulates. Attention measurements further show how frozen computation transforms Q/V translations into layer- and input-dependent effects. Together, these findings support selective plasticity as a design goal for unlabeled test-time learning: useful adaptation depends not only on optimizing the surrogate, but also on restricting the form of change it can induce.

REPRODUCIBILITY STATEMENT

We provide anonymous code for reproducing the main experiments at `https://anonymous.4open.science/r/review-code-FDE3/`. The paper and appendix specify the evaluated models and datasets, adaptation objectives, intervention locations, optimization settings, learning rates, training budgets, random seeds, and evaluation procedures used in our experiments. Additional tables report the full learning-rate sweeps, baseline configurations, and analysis settings needed to reproduce the reported comparisons.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, sentence-level rewriting, and organization of the manuscript. All scientific claims, experimental results, numerical values, methodological descriptions, and citations were checked and approved by the authors. The authors remain responsible for the accuracy, interpretation, and presentation of all content in the paper.

REFERENCES

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022. URL https://aclanthology.org/2022.acl-short.1/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

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

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Deryl Chua, Mak Lee, Gee Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/423d0909791493b7c10916fd328c2913-Abstract-Conference.html.

Bincheng Gu, Min Gao, Zongwei Wang, Yibing Bai, Yulan He, and Junliang Yu. Beyond Confidence: Stability-Aware Test-Time Adaptation for LLM Reasoning. *arXiv preprint arXiv:2609.11393*, 2026. URL https://arxiv.org/abs/2609.11393.

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. URL https://arxiv.org/abs/2103.03874.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24823–24849, 2025a. URL https://proceedings.mlr.press/v267/hu25z.html.

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. SLOT: Sample-specific language model optimization at test-time. *arXiv preprint arXiv:2505.12392*, 2025b. URL https://arxiv.org/abs/2505.12392.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. URL https://arxiv.org/abs/2206.14858.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2210.08823.

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

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. *arXiv preprint arXiv:2205.05638*, 2022. URL https://arxiv.org/abs/2205.05638.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3622–3628, 2020. doi: 10.24963/ijcai.2020/501. URL https://www.ijcai.org/proceedings/2020/501.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 16888–16905, 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of NAACL-HLT*, pp. 2080–2094, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Qwen Team. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.5-4B-Base model card. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B-Base.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

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. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.16049.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*, 2024. URL https://arxiv.org/abs/2410.12877.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. *arXiv preprint arXiv:1909.13231*, 2020. URL https://arxiv.org/abs/1909.13231.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Max Torop, Aria Masoomi, Masih Eskandar, and Jennifer Dy. DISCO: Disentangled communication steering for large language models. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-1938. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/53d60d0006475f877249ca7f520dc793-Abstract-Conference.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.10726.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract.html.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Advancing Parameter Efficiency in Fine-tuning via Representation Editing. *arXiv preprint arXiv:2402.15179*, 2024a. URL https://arxiv.org/abs/2402.15179.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation Finetuning for Language Models. *arXiv preprint arXiv:2404.03592*, 2024b. URL https://arxiv.org/abs/2404.03592.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. COME: Test-time adaption by conservatively minimizing entropy. *arXiv preprint arXiv:2410.10894*, 2024. URL https://arxiv.org/abs/2410.10894.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right Question is Already Half the Answer: Fully Unsupervised LLM Reasoning Incentivization. *arXiv preprint arXiv:2504.05812*, 2025. URL https://arxiv.org/abs/2504.05812.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2025. Hugging Face dataset, math-ai/aime25, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023. URL https://arxiv.org/abs/2304.06364.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning. *arXiv preprint arXiv:2504.16084*, 2025. URL https://arxiv.org/abs/2504.16084.

A Experimental Setup

A.1 Dataset Sizes

Table 2 lists the seven main evaluation tasks and their question counts. Both Qwen3-8B-Base and Qwen3.5-4B-Base are evaluated on the same 2,759 questions.

AIME25 comprises the 30 problems from the 2025 American Invitational Mathematics Examination (AIME I and II); we use the `math-ai/aime25` text release.

Table 2: Question counts for the seven main evaluation datasets.

Dataset	Questions
Minerva Math	272
LSAT-AR	230
MuSR Object Placements	256
ARC-Challenge	1,172
SVAMP	300
AIME25	30
MMLU-Pro Philosophy	499
Total	**2,759**

A.2 TTShift Adaptation Protocol

The final TTShift configuration learns zero-initialized, dataset-shared FP32 additive vectors at the Q/V projection outputs, while keeping the BF16 pretrained model frozen. The vectors are shared across all examples and token positions within a task and are reinitialized between tasks. The input next-token CE in Equation 3 averages the loss within each question and then over questions; gold answers, pseudo-labels, and generated solutions are not used for this objective. Task inputs retain the question, available choices, and fixed answer-format instruction; MMLU-Pro retains its self-contained task prompt.

Both model sizes use one transductive epoch with Adam at a fixed learning rate of 8×10^{-5}, betas $(0.9, 0.999)$, epsilon 10^{-8}, zero weight decay, microbatch 1, gradient accumulation 1, and gradient clipping at 1. There is no learning-rate scheduler or warmup. The seed is 20260920, and each task uses its fixed shuffled question order. The optimizer and shifts reset between tasks. Evaluation retains the final checkpoint without accuracy-based early stopping or task-specific checkpoint selection.

Learning-rate selection. The fixed TTShift learning rate of 8×10^{-5} was selected on GPQA Diamond and then applied unchanged to both models and all seven main evaluation tasks. GPQA Diamond is the learning-rate selection task; the main-task sweep measures stability and is not used to select a new learning rate. The main-table LoRA baseline uses 10^{-6}, following the learning-rate choice for ReasoningBench in TLM (Hu et al., 2025a) and selected on GPQA Diamond before being applied unchanged to both models and all seven main evaluation tasks. Neither method uses task-specific learning-rate tuning for the main table. GPQA Diamond is used only for method-development decisions (shift placement and learning-rate selection) and is excluded from the seven-task main evaluation.

On Qwen3-8B-Base, shifts are learned at 72 sites across all 36 layers: 36 Q outputs of width 4,096 and 36 V outputs of width 1,024, giving $36(4096{+}1024) = 184{,}320$ trainable parameters. Qwen3.5-4B-Base uses the Q/V projections in its eight full-attention layers, with 16 sites and 73,728 parameters; the remaining hybrid blocks are not adapted by this configuration.

A.3 Baseline Reproduction

We reproduce LoRA, TLM, SLOT, and EATA on the unlabeled target inputs of each task. Gold answers are used only for evaluation. Each method is evaluated with one greedy completion per

question and a maximum of 4,096 generated tokens; task-specific answer extraction and scoring are shared with Base.

LoRA. LoRA (Hu et al., 2022) learns low-rank increments to the Q/V projection weights while freezing all pretrained parameters. For both models, the main-table results use rank 8, $\alpha = 16$, zero dropout, and one transductive epoch of unlabeled input next-token CE with Adam at 10^{-6}. Adam uses betas $(0.9, 0.999)$, epsilon 10^{-8}, and zero weight decay. The microbatch size and gradient accumulation are both 1, gradient clipping is 1, and the seed is 20260920; there is no scheduler or warmup. Qwen3-8B-Base adapts Q/V in all 36 layers, while Qwen3.5-4B-Base adapts Q/V in its eight full-attention layers. Each task uses its fixed shuffled question order, and evaluation retains the final one-epoch checkpoint.

TLM. TLM (Hu et al., 2025a) adapts rank-8 q/v LoRA with $\alpha = 16$ and zero dropout through filtered, weighted input CE. For input i, let c_i^0 denote its frozen-base CE and c_i its adapted-model CE. We use the weight $w_i = 0.1 \exp(c_i^0 - 3)\mathbf{1}[c_i^0 > 3]$ and optimize the weighted CE over selected inputs. Training runs for one epoch with AdamW at 10^{-6}, betas $(0.9, 0.999)$, epsilon 10^{-8}, microbatch 1, gradient accumulation 1, cosine learning-rate decay, 10% warmup, zero weight decay, gradient clipping at 1, and seed 42.

SLOT. SLOT (Hu et al., 2025b) freezes the language model and optimizes a zero-initialized additive vector at the final hidden representation for each question. The vector is trained on that question's prompt next-token CE for three AdamW steps and retained for its answer generation. We use learning rate 0.01, betas $(0.9, 0.999)$, epsilon 10^{-5}, and weight decay 10^{-8}. A fresh vector is initialized for every question.

EATA. EATA (Niu et al., 2022) uses rank-8 q/v LoRA with $\alpha = 16$ and zero dropout. For each question, it greedily generates up to eight tokens and averages their prediction entropies to obtain H. Questions with $H < 0.4$ pass the reliability filter. The mean token predictive distribution is compared with a running distribution using an absolute cosine-similarity threshold of 0.05; accepted distributions update the running average with coefficient 0.9. Accepted questions minimize $H \exp(0.4 - \mathrm{stopgrad}(H))$, and rejected questions skip the optimizer step. We use AdamW at 5×10^{-6}, betas $(0.9, 0.999)$, epsilon 10^{-8}, microbatch 1, gradient accumulation 1, one epoch, cosine decay, 10% warmup, zero weight decay, gradient clipping at 1, and seed 42. This is a language-model transfer with question-level aggregation, rather than the original vision adaptation procedure; no Fisher regularizer is used.

B Shared-Shift Formation and Norm Robustness

B.1 Measurement Protocol

The formation analysis in Section 4.4 covers all seven main tasks on Qwen3-8B-Base using the final Q/V TTSHIFT recipe in Appendix A.2: Adam at 8×10^{-5}, one unlabeled input-CE epoch, and the fixed seed and question order. We save 0/10/25/50/75/100% checkpoints, including zero initialization, without choosing checkpoints by answer accuracy. For Minerva Math, the saved steps are 0/28/68/136/204/272; for LSAT-AR they are 0/23/58/115/173/230.

At each checkpoint, a read-only pass computes each question's CE gradient with respect to all 36 Q and 36 V shifts. No optimizer update is performed during this measurement. Corresponding tensors are flattened across layers into Q, V, or joint-QV vectors. The saved analysis retains per-layer gradients, both norm definitions below, and leave-one-out cosine distributions.

B.2 Gradient Agreement and Shift Trajectories

For example i and component $m \in \{Q, V, QV\}$, let g_i^m denote the flattened CE gradient. We measure normalized gradient consensus as

$$C_p^m = \frac{\|\sum_i g_i^m\|_p}{\sum_i \|g_i^m\|_p}, \qquad p \in \{1, 2\}. \tag{5}$$

The main analysis uses L_2 consensus; L_1 is a complementary norm-robustness check. Leave-one-out (LOO) alignment is the cosine between g_i^m and $\sum_{j \neq i} g_j^m$. The positive fraction counts valid cosines strictly above zero; zero-norm vectors do not define a cosine.

Let $\beta^{(s)}$ be the flattened joint Q/V shift at a nonzero progress checkpoint s and $\beta^{(100)}$ its final value. Directional similarity is $\cos(\beta^{(s)}, \beta^{(100)})$, and the fraction of final norm is $\|\beta^{(s)}\|_2 / \|\beta^{(100)}\|_2$. These quantities distinguish changes in direction from the accumulation of magnitude. The initial zero shift has zero norm and no defined directional cosine. Across seven tasks, the median cosine at 50% progress is 0.916, while the median fraction of final norm is 62.5%, as reported in the main text.

B.3 Norm Robustness

Table 3 reports Q- and V-specific consensus under both norm definitions for Minerva Math and LSAT-AR at every predefined checkpoint. These two tasks are examined in detail here; the main formation figure covers all seven tasks.

Table 3: Gradient consensus across all predefined checkpoints. $C_p^m = \|\sum_i g_i^m\|_p / \sum_i \|g_i^m\|_p$; LOO+ is the percentage of per-question gradients with positive cosine to the leave-one-out aggregate. Q and V are flattened over all 36 layers. These are descriptive input-CE gradient statistics, not accuracy-based checkpoint selections.

Task	Progress	C_1^Q	C_2^Q	Q LOO+	C_1^V	C_2^V	V LOO+
Minerva Math	0%	0.218	0.184	92.3	0.298	0.276	99.6
	10%	0.238	0.201	87.9	0.322	0.300	99.3
	25%	0.262	0.304	66.5	0.272	0.261	96.0
	50%	0.213	0.249	66.5	0.224	0.206	93.4
	75%	0.159	0.124	77.6	0.222	0.193	98.2
	100%	0.256	0.436	75.0	0.229	0.225	97.4
LSAT-AR	0%	0.342	0.304	94.8	0.463	0.436	100.0
	10%	0.306	0.257	91.7	0.414	0.362	100.0
	25%	0.290	0.256	93.5	0.361	0.294	100.0
	50%	0.290	0.234	96.5	0.367	0.302	100.0
	75%	0.253	0.199	98.3	0.322	0.273	100.0
	100%	0.277	0.198	100.0	0.369	0.311	100.0

Evaluation and interpretation. Checkpoint evaluation uses an independently initialized vLLM process per checkpoint to isolate prefix caches, with input prompts capped at 4,096 tokens, generations at 4,096 tokens, and context at 8,192 tokens. Each task retains its prompt and scorer, and invalid answers count as incorrect. Repair, damage, and accuracy changes are measured against the same task's isolated zero-initialization checkpoint. The seven-task analysis contains 42 checkpoint records and 16,554 item/checkpoint comparisons. Prediction flips compare extracted answers, including an invalid-answer category, and need not equal repairs plus damage. Shared formatting can contribute to gradient agreement. These single-seed, transductive measurements describe observed trajectories; they do not establish that gradient agreement, direction alignment, or norm growth causes downstream gains or degradation.