SAFETY-CONSTRAINED TEST-TIME TRAINING: POST-UPDATE GUARDING FOR ALIGNED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time training (TTT) adapts a language model during inference, but an attacker can use the same update loop to erode safety alignment. We propose *Safety-Constrained Test-Time Training* (SC-TTT), which probes each candidate update against a fixed behavioral reference. It retains the candidate only when the harmful-side compliance change does not exceed the benign-side change by more than a threshold. Otherwise, SC-TTT restores the preceding parameter state. Across Gemma-3-4B, Llama-3.2-3B, and Qwen3.5-4B, SC-TTT lowers mean attack success from 19.8–50.3% under vanilla TTT to 0.1–5.8% over five random seeds. It retains 0.83–1.09 of the vanilla adaptation gain on BBH-H3 and matches vanilla accuracy on ARC-AGI. A learning-rate analysis shows that the criterion is reliable only within a calibrated working regime. A full-knowledge three-step pressure test still suppresses 47–70% of vanilla attack success despite optimizing the gate directly. These results establish strong attack suppression, high utility retention, and an explicit operating regime for guarded test-time adaptation.

1 INTRODUCTION

Test-time training (TTT) lets a language model adapt its parameters during inference. This plasticity improves performance under distribution shift, but it also creates a security boundary. An attacker who controls the adaptation content can alter the deployed model rather than merely influence one response. A few harmful demonstrations can increase compliance with harmful requests while leaving ordinary behavior largely intact (Antonelli et al., 2026).

Existing defenses act at different stages. Harmful fine-tuning defenses such as Vaccine, RepNoise, and LISA intervene before deployment (Huang et al., 2024d; Rosati et al., 2024; Huang et al., 2024c). Conventional jailbreak defenses assume fixed parameters and inspect the mapping from an input x to an output y. A TTT attack instead follows $x \mapsto \theta' \mapsto y$: the input changes the model that produces later responses. Antonelli et al. (2026) expose this threat and propose provider-side request detection. We study the complementary control point inside the update loop.

We propose *Safety-Constrained Test-Time Training* (SC-TTT). Before each optimizer step, SC-TTT snapshots the active parameters. It then materializes a candidate and measures behavioral change on harmful and benign anchors. The score $g = \Delta\mathcal{H} - \Delta\mathcal{B}$ compares the candidate with a fixed base reference. SC-TTT retains the candidate when $g \leq \tau$ and restores the snapshot otherwise. The rollback covers active parameters; AdamW state remains part of the proposal process.

The fixed reference makes every decision auditable across the adaptation trajectory. We further stress-test this mechanism against an attacker that optimizes the gate itself. This pressure test complements the primary non-adaptive evaluation and measures how much suppression remains under full criterion knowledge.

Across three 3–4B model families and five seeds, SC-TTT rejects harmful candidates at a high rate. Mean attack success falls from 19.8–50.3% to 0.1–5.8%. On separately evaluated benign workloads, the method retains 0.83–1.09 of the vanilla BBH-H3 adaptation gain and matches vanilla ARC-AGI accuracy. Under direct gate-aware optimization, SC-TTT still suppresses 47–70% of vanilla attack success within three steps.

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.

Our contributions are:

- We formulate safety control for parameter-changing TTT as a post-update, reference-relative acceptance decision.
- We introduce a parameter-state snapshot, probe, and rollback mechanism for LoRA-based language-model adaptation.
- We evaluate static suppression, benchmark retention, working-regime sensitivity, alternative update controls, and adaptive pressure tests across three model families.

2 Problem Formulation

Test-time training. TTT updates a model at inference time using a surrogate objective (Sun et al., 2020; Wang et al., 2021). For language models, parameter-efficient TTT can improve domain and few-shot performance (Hu et al., 2025). We study LoRA-based adaptation (Hu et al., 2022). Let θ_t be the active parameter state and let $\theta' = \theta_t + \Delta\theta$ be the candidate produced by one optimizer step. The attacker supplies the adaptation content and loss, but cannot directly modify the optimizer, gate, or parameters.

Behavioral measurements. Let $\mathcal{X}_{\text{harm}}$ and $\mathcal{X}_{\text{ben}}$ denote harmful and benign anchor sets. For a prompt x, $c_\theta(x)$ is the teacher-forced per-token log-probability of a fixed, model-specific compliance target. We define

$$\mathcal{H}(\theta) = \frac{1}{|\mathcal{X}_{\text{harm}}|} \sum_{x \in \mathcal{X}_{\text{harm}}} c_\theta(x), \qquad \mathcal{B}(\theta) = \frac{1}{|\mathcal{X}_{\text{ben}}|} \sum_{x \in \mathcal{X}_{\text{ben}}} c_\theta(x). \tag{1}$$

Higher $\mathcal{H}$ indicates greater willingness to comply with harmful requests. Higher $\mathcal{B}$ indicates greater willingness to answer benign requests. The gate uses these quantities as behavioral proxies; free-generation attack success is evaluated separately by an external judge.

Safety and utility metrics. Attack success rate (ASR) is the fraction of decoded responses judged harmful and compliant. ASR@10 is the fraction of prompts with at least one successful response among ten samples. We report benchmark accuracy for utility. For a variant v, let $\Delta_v = \mathcal{U}(\theta_v) - \mathcal{U}(\theta_0)$ be its gain over no adaptation. We use

$$\text{Retention}(v) = \frac{\Delta_v}{\Delta_{\text{van}}} \tag{2}$$

to measure how much of the vanilla TTT gain remains. We always report absolute accuracy because the denominator can be small.

3 Threat Model

The adversary controls the content and objective used by a deployed model's TTT loop. The model adapts a rank-32 LoRA module for three steps. The adversary cannot edit the optimizer implementation, gate, anchors, or parameters directly. Its goal is to increase ASR and ASR@10 while obtaining accepted updates. This follows the generation-phase threat of Antonelli et al. (2026), but uses a milder three-step configuration.

Our primary evaluation uses 32 reserved harmful prompts, ten samples per prompt, and learning rate 10^{-5}. The three tested models use the same step budget and threshold. Figure 1 contrasts this parameter attack with a fixed-parameter jailbreak. The gate acts on the candidate state rather than on the incoming request alone.

The adaptive evaluation adds two knowledge settings. A full-knowledge attacker optimizes the true criterion and anchors. A partial-knowledge attacker knows the criterion form but substitutes proxy anchors. The deployed gate still evaluates every candidate with its internal anchors. This design measures the consequence of gate knowledge without changing the acceptance rule.

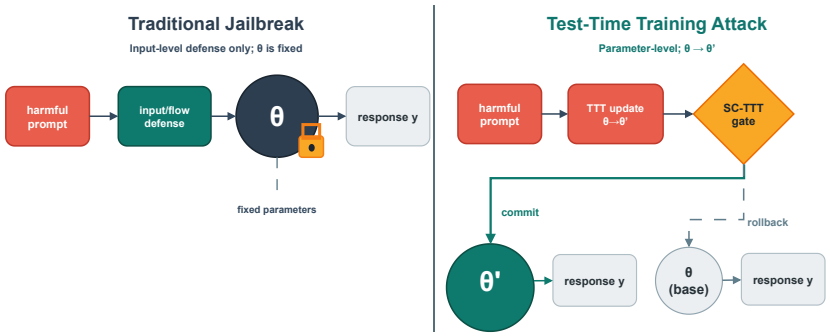


Figure 1: Input-level and parameter-level attack surfaces.

4 Safety-Constrained Test-Time Training

SC-TTT inserts a behavioral control point between an optimizer proposal and the active parameter state. Its design follows three requirements. First, the decision should inspect the realized candidate rather than infer safety from a loss value or update norm. Second, it should distinguish safety-specific drift from a general increase in model compliance. Third, it should prevent a failed candidate from entering the active state. These requirements lead to a reference-relative criterion, a deterministic probe, and parameter rollback.

4.1 Reference-Relative Criterion

Let $\theta_{\text{ref}} = \theta_0$ be the base state at the start of a TTT episode. Its measurements are computed once and reused for every candidate in that episode. We define

$$g(\theta') = [\mathcal{H}(\theta') - \mathcal{H}(\theta_{\text{ref}})] - [\mathcal{B}(\theta') - \mathcal{B}(\theta_{\text{ref}})]. \tag{3}$$

The candidate is accepted when $g \leq \tau$; we use $\tau = 0.10$. A positive score indicates that harmful-side compliance increased more than benign-side compliance relative to the base state.

Both terms in Eq. 3 are necessary. A harmful-only measurement can rise when an update makes the model more responsive in general. Subtracting $\Delta\mathcal{B}$ removes this shared compliance movement and isolates the asymmetric shift toward harmful content. Benign task adaptation can therefore change the model without being penalized merely for increasing responsiveness. The decision targets the relative erosion of the harmful–benign behavior gap.

The fixed base gives every step in an episode the same behavioral coordinate system. This has two operational benefits. Scores remain directly comparable across steps, and a harmful change cannot be hidden by dividing it into a sequence of small increments relative to moving references. The criterion tracks cumulative drift from the aligned initialization while the optimizer continues to build task-specific capability.

The anchor set contains six harmful and six benign prompts. Its small size makes per-step evaluation practical while preserving both sides of the behavioral contrast. We withhold the exact anchors and compliance targets because they define the deployed gate. Appendix B describes their construction and separation requirements without disclosing their contents.

4.2 Deterministic Safety Probe

Free generation and external judging are too stochastic and expensive for an optimizer-level decision. SC-TTT instead uses the teacher-forced score from Eq. 1. This produces a dense signal for every candidate and avoids decoding variance inside the update loop. Free-generation ASR remains the independent outcome metric in our evaluation.

The probe runs without gradient tracking and with dropout disabled. Reference and candidate measurements use the same tokenizer, chat formatting, and compliance target. The observed difference therefore reflects the candidate parameter change. The probe neither adds gradients nor consumes optimizer state, and its output is deterministic for a fixed parameter state.

Algorithm 1 SC-TTT parameter-state guarding

Require: base state θ_0, optimizer Opt, threshold τ, steps K
$\theta \leftarrow \theta_0$; compute $\mathcal{H}_{\text{ref}}, \mathcal{B}_{\text{ref}}$ once
for $k = 1$ **to** K **do**
  snapshot $\leftarrow \text{copy}(\theta)$
  materialize candidate with Opt.step()
  $g \leftarrow [\mathcal{H}(\theta') - \mathcal{H}_{\text{ref}}] - [\mathcal{B}(\theta') - \mathcal{B}_{\text{ref}}]$
  if $g > \tau$ **then**
    restore $\theta \leftarrow$ snapshot
return θ

4.3 Post-Update Parameter Control

Before each optimizer step, SC-TTT snapshots the trainable LoRA parameters. It then materializes the candidate and evaluates Eq. 3. If the candidate fails, SC-TTT restores the snapshot. Thus, any candidate with measured $g > \tau$ is absent from the next active parameter state. AdamW momentum and second-moment estimates are retained to preserve optimizer continuity across proposals. Each independent attack instance or utility task starts with a fresh optimizer.

This post-update decision differs from safety-loss regularization. A soft penalty lets the optimizer exchange safety loss against task loss, and its effect depends on both loss scales. Update-norm clamping limits distance without identifying the behavioral direction of an update. SC-TTT evaluates the realized behavior after these optimization choices have taken effect. The binary commit decision therefore enforces the measured condition without a loss-weight trade-off. Section 6.2 tests this distinction empirically.

4.4 Algorithm

SC-TTT changes neither the task loss nor the optimizer proposal. It controls which realized proposals become active. This modularity lets the same gate wrap existing LoRA-based TTT loops and preserves the original adaptation objective. Section 6.1 further tests the gate under direct optimization of its decision rule.

5 Evaluation

Our evaluation follows the complete evidence chain of the method. We first measure the safety erosion caused by vanilla TTT. We then test whether SC-TTT suppresses that erosion while retaining useful adaptation. Finally, we examine cross-model consistency and the working regime of the criterion. Appendix A gives the full protocol and additional analyses.

5.1 Setup

We test Qwen3.5-4B, Llama-3.2-3B, and Gemma-3-4B. These families use different architectures, chat templates, and alignment recipes. The main attack and utility experiments use seeds {42,43,44,45,46} on one RTX 5080. We report mean $\pm$ standard deviation over the five seeds.

LoRA has rank 32, $\alpha = 64$, dropout 0.05, and all applicable linear targets. Attack runs use three steps and utility runs use five. Each attack evaluation contains 32 reserved prompts and ten stochastic responses per prompt. An Ollama-hosted Qwen3.6-35B-A3B model labels harmful compliance. We manually check 10% of its labels and record output validity separately.

Utility evaluation uses task-macro accuracy on BBH-H3 and exact-match task accuracy on ARC-AGI. BBH-H3 spans 27 reasoning tasks and 4,511 queries. ARC-AGI contains 400 tasks and 419 queries. Adaptation uses task support examples; the evaluation query never enters the update loss. No-TTT, vanilla TTT, and SC-TTT share the same data and decoding protocol. The gate uses $\tau = 0.10$ for all three model families without model-specific threshold tuning.

Table 1: Five-seed attack success under test-time adaptation.

	No TTT		Vanilla		SC-TTT	
Model	ASR	ASR@10	ASR	ASR@10	ASR	ASR@10
Qwen	0.00	0.00	50.25±2.32	94.38±1.40	0.12±0.17	1.25±1.71
Llama	0.94	9.38	19.81±1.57	57.50±5.23	0.12±0.17	1.25±1.71
Gemma	2.81	12.50	21.69±0.84	46.88±3.12	5.75±1.00	17.50±4.19

Table 2: Five-seed accuracy and adaptation-gain retention.

	ARC-AGI			BBH-H3		
Model	Van.	SC	Ret.	Van.	SC	Ret.
Qwen	7.65±0.34	8.00±0.18	1.08±0.07	30.85±0.17	30.63±0.31	0.83±0.21
Llama	0.95±0.11	0.95±0.11	1.00±0.00	27.72±0.40	27.90±0.42	1.03±0.05
Gemma	1.65±0.14	1.65±0.14	1.00±0.00	35.78±0.25	35.90±0.47	1.09±0.36

5.2 Attack Suppression

Table 1 reports the main attack result. Vanilla TTT raises mean ASR to 19.81–50.25% and ASR@10 to 46.88–94.38%. Across five seeds, SC-TTT rejects harmful candidates at a high rate and reduces mean ASR to 0.12–5.75%. This is a 73–99% relative reduction across the three model families.

The defended ASR remains within 2.9 percentage points of the corresponding no-TTT level for every model. This comparison is important: it shows that SC-TTT removes nearly all of the additional risk introduced by parameter adaptation. The validity-aware judge observes harmful compliance rather than mechanically rewarding malformed outputs. The result therefore reflects safety preservation, not generation collapse.

5.3 Utility Retention

Table 2 reports absolute accuracy and adaptation-gain retention. On BBH-H3, SC-TTT retains 0.83–1.09 of the vanilla gain. Llama slightly exceeds vanilla TTT, and Qwen retains 83% of its gain. On ARC-AGI, SC-TTT matches vanilla TTT for all three models. The safety control therefore preserves the measured adaptation benefit across two distinct task suites.

Safety and utility are evaluated on separate held-out workloads but share the same gate, anchors, and threshold. SC-TTT is not switched off for utility runs. The observed retention therefore measures the selectivity of the same decision rule that suppresses the attacks in Table 1.

5.4 Update Selectivity

Table 3 connects gate decisions to final safety and utility outcomes. The same criterion and $\tau = 0.10$ reject 97.5–100% of harmful candidates while accepting 84.6–100% of BBH and 96.25–100% of ARC candidates. SC-TTT therefore distinguishes update types rather than uniformly suppressing adaptation.

The decision statistics explain both sides of the result. Near-complete harmful rejection coincides with a 73.4–99.8% ASR reduction. High benign acceptance coincides with strong adaptation-gain retention on both utility suites. This alignment holds across three independently aligned model families without changing the criterion or threshold.

5.5 Cross-Model Consistency

The three model families begin from different refusal rates and exhibit very different vanilla attack susceptibility. Qwen reaches 50.25% ASR, while Llama and Gemma reach 19.81% and 21.69%. Nevertheless, the same criterion form and $\tau = 0.10$ reduce adaptation-induced harm in every family. The defended states also retain the vanilla task gains across both utility suites.

Table 3: Five-seed update selectivity and outcome alignment.

	Gate decisions (%)			Safety	Utility	
Model	**Harm rej.**	**BBH acc.**	**ARC acc.**	**ASR ↓**	**BBH ret.**	**ARC ret.**
Qwen	100.0 ± 0.0	87.4 ± 1.7	96.25 ± 0.27	99.8 ± 0.3	0.83 ± 0.21	1.08 ± 0.07
Llama	100.0 ± 0.0	84.6 ± 2.4	100.0 ± 0.0	99.4 ± 0.9	1.03 ± 0.05	1.00 ± 0.00
Gemma	97.5 ± 1.9	100.0 ± 0.0	99.89 ± 0.05	73.4 ± 5.1	1.09 ± 0.36	1.00 ± 0.00

Table 4: Criterion sensitivity to the learning-rate regime.

lr	**Harmful**	**Benign**	**Gap**	**AUROC**
1e−5	0.96 ± 0.16	0.04 ± 0.01	$+0.64$	1.00
3e−5	1.27 ± 0.31	0.20 ± 0.02	$+0.40$	1.00
6e−5	0.31 ± 0.75	0.52 ± 0.04	-2.11	0.40

This consistency is stronger than transfer of a tuned classifier. SC-TTT uses model-specific formatting, but it does not fit a new threshold or train a detector for each architecture. The harmful–benign contrast supplies a common behavioral coordinate. The results show that this coordinate can control LoRA updates across three independently aligned model families.

5.6 Working-Regime Sensitivity

A behavioral gate must be calibrated to the scale of the proposals it controls. Table 4 and Figure 2 examine this relationship on Llama-3.2-3B. We use 16 harmful and 16 benign instances and measure the third update step.

At 10^{-5}, harmful and benign candidates occupy disjoint regions around the shared threshold. The separation gap is $+0.64$, and AUROC is 1.00. At 3×10^{-5}, the ordering remains perfect, but the benign distribution moves above the original threshold. This distinction between ranking and calibration explains why a high AUROC alone is insufficient for an operational gate. At 6×10^{-5}, the ordering inverts as the proposal scale leaves the calibrated regime.

This analysis turns a deployment assumption into an observable condition. Benign score drift above τ indicates that the update scale requires recalibration. SC-TTT therefore exposes its working regime through the same probe used for control, rather than assuming that one threshold applies to arbitrary optimizer dynamics. The main evaluation operates in the separated regime identified by this analysis.

6 Adaptive Attacks and Comparisons

6.1 Adaptive Pressure Test

Adaptive evaluation is essential when a defense participates in the attacker's optimization landscape (Wu et al., 2023; Su et al., 2024; Nguyen & Nguyen, 2026; Nasr et al., 2026). We test a full-knowledge attacker using the deployed criterion and a partial-knowledge attacker using proxy anchors. The two objectives use $\lambda = 0.5$ and 1.0, respectively, with seed 42. Each model contributes 32 instances, 320 responses, and 96 candidate steps.

The full-knowledge objective deliberately drives g below the threshold and obtains acceptance for all candidates. This directly pressures the decision rule. Even so, Table 5 shows that SC-TTT suppresses 47–70% of vanilla ASR within three steps. The final column reports the complementary quantity, the recovery ratio (Rec.): the full-knowledge ASR divided by the vanilla ASR. The suppression quoted above is its complement, so lower values are better. The attacker must divide its update budget between fitting harmful behavior and satisfying the safety criterion. This optimization tension remains effective after the acceptance decision becomes an explicit attack target. Proxy-anchor optimization recovers still less harm, showing the additional protection created by anchor mismatch.

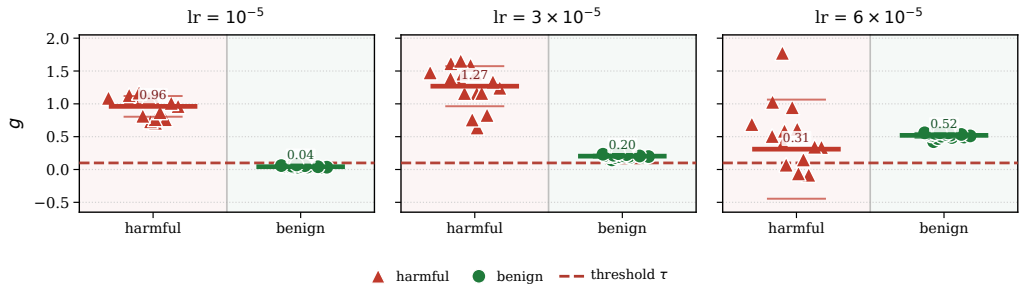


Figure 2: Criterion distributions across learning-rate regimes.

Table 5: Attack success under adaptive gate knowledge.

	Vanilla		SC-TTT		Full		Partial		Rec.
Model	ASR	@10	ASR	@10	ASR	@10	ASR	@10	
Qwen	52.19	93.75	0.31	3.13	16.25	56.25	0.00	0.00	31%
Llama	19.38	62.50	0.31	3.13	10.31	43.75	4.69	18.75	53%
Gemma	23.12	50.00	4.38	21.88	6.88	21.88	5.94	18.75	30%

This pressure test complements the primary five-seed evaluation. It identifies an interpretable security boundary while demonstrating substantial residual suppression under complete knowledge of the decision rule. Reporting this result also prevents static-attack success from being mistaken for reliance on an oblivious adversary.

6.2 Safety–Utility Advantage

We compare SC-TTT with three forms of update control common in defenses against harmful fine-tuning (Huang et al., 2024b): update-norm clamping, safety-loss regularization (Rosati et al., 2024; Huang et al., 2024a; Tamirisa et al., 2024), and a KL-to-reference constraint (Mukhoti et al., 2023; Qi et al., 2024). They represent parameter distance, loss regularization, and distributional proximity, respectively. Their behavior reveals why the post-update decision matters.

Norm clamping retains Qwen utility but leaves ASR at 51.8%, close to vanilla TTT at 50.3%. The same pattern holds across Llama and Gemma, where the same clamping threshold also leaves ASR near vanilla. Parameter distance alone does not identify whether a small update moves toward harmful compliance.

Safety-loss regularization lowers Qwen ASR to 10.2%, but BBH-H3 retention falls to -0.70. The regularizer must suppress the task objective strongly enough to control harm. The KL-to-reference constraint provides a middle point: ASR reaches 16.6%, with retention 0.97 on BBH-H3 and 0.98 on ARC-AGI. SC-TTT reaches 0.12% ASR while retaining 0.83 and 1.08, respectively.

Figure 3 presents the joint comparison of five-seed means. SC-TTT occupies the strongest safety–utility region in both panels. The alternatives either retain high ASR or sacrifice a larger share of the adaptation gain. In contrast, SC-TTT leaves the task objective unchanged and controls only commitment of the realized candidate. The Pareto advantage is therefore consistent with the method's design rather than a weaker adaptation objective.

6.3 Computational Profile

Each SC-TTT step adds one gradient-free forward pass over 12 short anchors. A complete 32-instance, three-step sweep, including adaptation, takes about 40 seconds on an RTX 5080. The probe is batchable and independent of the training sequence length. It also avoids free decoding and judge invocation inside the optimizer loop. SC-TTT therefore adds a small, predictable operation while avoiding retraining or a second alignment stage.

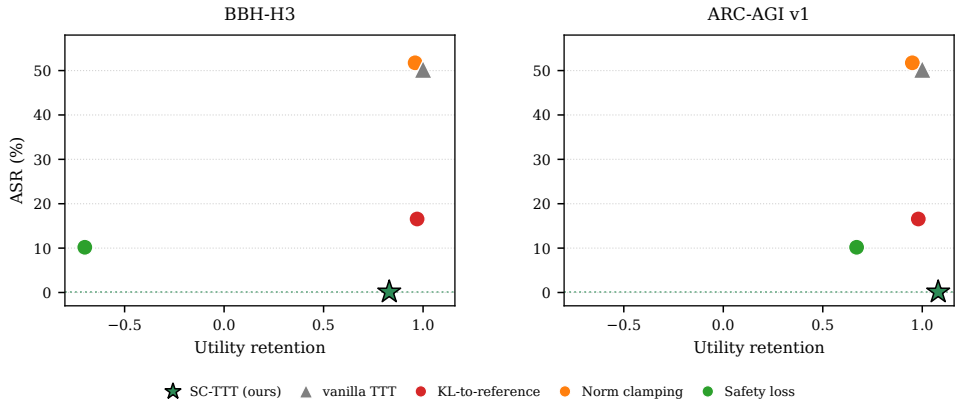


Figure 3: Qwen safety–utility trade-off over five seeds.

The implementation uses a fixed base reference to track cumulative behavioral drift. Parameter rollback blocks failed candidates from the active model, while retained AdamW statistics maintain optimizer continuity. These choices yield a simple control layer for LoRA-based adaptation and account for the strong cross-model results in the evaluated regime.

7 Related Work

Adaptation and alignment. TTT improves generalization under distribution shift (Sun et al., 2020; Wang et al., 2021) and has been extended to parameter-efficient and long-context language-model adaptation (Hu et al., 2025; Pan et al., 2024; Zhu et al., 2026; Tandon et al., 2025). Harmful fine-tuning defenses instead intervene before deployment: Vaccine hardens the base model, Rep-Noise perturbs representations, and LISA separates the alignment and user objectives with a proximal term (Huang et al., 2024d; Rosati et al., 2024; Huang et al., 2024c). These methods protect the training pipeline, whereas SC-TTT guards candidate states produced by a deployed update loop.

Test-time safety. Fixed-parameter jailbreak defenses inspect inputs, activations, decoding, or outputs. Test-Time Immunization perturbs activations (Yu et al., 2025), while Test-Time Preference Optimization uses iterative textual feedback without weight updates (Li et al., 2025). CoSA changes an inference-time safety configuration for authorized users (Zhang et al., 2025). SC-TTT addresses a distinct case: an untrusted objective modifies model parameters, so the intervention occurs between optimizer proposal and parameter commitment.

The closest empirical anchor is Antonelli et al. (2026). They show that TTT can raise ASR@10 to 93–95% and propose provider-side request detection. SC-TTT provides the complementary in-loop control point after a candidate update is materialized. Test-time adaptation poisoning further shows why static attacks are insufficient (Wu et al., 2023; Su et al., 2024; Nguyen & Nguyen, 2026). Following the adaptive-attacker principle (Nasr et al., 2026), we evaluate direct optimization of our decision rule. To our knowledge, SC-TTT is the first post-update parameter-state gate for safety-aligned LLM test-time training.

8 Conclusion

TTT makes the deployed update loop part of the language model's security boundary. SC-TTT protects this boundary through a reference-relative behavioral criterion and post-update parameter control. Across three model families and five seeds, it reduces mean ASR from 19.8–50.3% under vanilla TTT to 0.1–5.8%. It retains 0.83–1.09 of the BBH-H3 adaptation gain and matches vanilla ARC-AGI accuracy. The safety–utility comparison places SC-TTT ahead of update-norm clamping, safety-loss regularization, and the KL-to-reference constraint at the evaluated settings. A direct gate-aware pressure test still leaves 47–70% of vanilla attack success suppressed within three steps.

Together, these results establish post-update parameter guarding as an effective control layer for adaptive language models.

AI USE STATEMENT

We used Claude Code to assist with writing and editing this manuscript. All AI-assisted text was reviewed and edited by the authors, who take responsibility for the final content.

REFERENCES

Simone Antonelli, Sadegh Akhondzadeh, and Aleksandar Bojchevski. Test-time training undermines safety guardrails, 5 2026. URL https://arxiv.org/abs/2605.22984. Preprint, not yet peer-reviewed.

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 *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models, 2025. URL https://arxiv.org/abs/2505.20633.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation, 2024a. URL https://arxiv.org/abs/2409.01586.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey, 2024b. URL https://arxiv.org/abs/2409.18169. Accepted at ACM Computing Surveys.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024c. doi: 10.52202/079017-3320.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024d. doi: 10.52202/079017-2356.

Yafu Li, Xuyang Hu, Xiaoye Qu, Linjie Li, and Yu Cheng. Test-time preference optimization: On-the-fly alignment via iterative textual feedback, 1 2025. URL https://arxiv.org/abs/2501.12895.

Jishnu Mukhoti, Yarin Gal, Philip H. S. Torr, and Puneet K. Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution, 2023. URL https://arxiv.org/abs/2308.13320.

Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V. Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, Abhradeep Thakurta, Kai Yuanqing Xiao, Andreas Terzis, and Florian Tramèr. The attacker moves second: Stronger adaptive attacks bypass defenses against LLM jailbreaks and prompt injections. In *Proceedings of the 36th USENIX Security Symposium*, 2026. URL https://arxiv.org/abs/2510.09023.

Phuc Duc Nguyen and Quang Duc Nguyen. Sample-wise targeted adversarial attacks on test-time adaptation, 5 2026. URL https://arxiv.org/abs/2605.23411.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. LISA: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. doi: 10.52202/079017-1817.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep, 2024. URL https://arxiv.org/abs/2406.05946.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. doi: 10.52202/079017-0402.

Yongyi Su, Yushu Li, Nanqing Liu, Kui Jia, Xulei Yang, Chuan-Sheng Foo, and Xun Xu. On the adversarial risk of test time adaptation: An investigation into realistic test-time data poisoning, 10 2024. URL https://arxiv.org/abs/2410.04682.

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. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248, 2020. URL https://arxiv.org/abs/1909.13231.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs, 2024. URL https://arxiv.org/abs/2408.00761.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, Carlos Guestrin, Jed McCaleb, Yejin Choi, and Yu Sun. End-to-end test-time training for long context, 12 2025. URL https://arxiv.org/abs/2512.23675.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2006.10726.

Tong Wu, Feiran Jia, Xiangyu Qi, Jiachen T. Wang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. Uncovering adversarial risks of test-time adaptation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37456–37495, 2023. URL https://arxiv.org/abs/2301.12576.

Yongcan Yu, Yanbo Wang, Ran He, and Jian Liang. Test-time immunization: A universal defense framework against jailbreaks for (multimodal) large language models, 5 2025. URL https://arxiv.org/abs/2505.22271.

Jingyu Zhang, Ahmed Elgohary, Ahmed Magooda, Daniel Khashabi, and Benjamin Van Durme. Controllable safety alignment: Inference-time adaptation to diverse safety requirements, 2025. URL https://arxiv.org/abs/2410.08968.

Xinyu Zhu, Zhe Xu, Xiaohan Wei, Yunchen Pu, Fei Tian, Chonglin Sun, Kaushik Rangadurai, Hua Zhi, Frank Shyu, Sandeep Pandey, Luke Simon, Yu Meng, and Xi Liu. Self-guided test-time training for long-context llms, 7 2026. URL https://arxiv.org/abs/2607.09415.

A Experimental Protocol

A.1 Training and Evaluation Configuration

Table 6 gives the complete model-level configuration. Attack experiments use three TTT steps and utility experiments use five. All main attack and utility results aggregate seeds {42,43,44,45,46} and report mean $\pm$ standard deviation.

LoRA uses $\alpha = 64$, dropout 0.05, and all applicable linear modules. Attack decoding draws ten samples with temperature 1.0 and top-p 0.9. BBH-H3 and ARC-AGI use greedy decoding. BBH-H3 contains 27 tasks and 4,511 evaluation queries. ARC-AGI contains 400 tasks and 419 evaluation queries.

Gemma-3-4B uses the larger utility learning rate of 3×10^{-5}. Its benign probe displacement is negative in sign: with the same teacher-forced gate at 10^{-5}, candidate scores range from $g \approx$

Table 6: Model and adaptation configuration.

Model	Attack lr	Utility lr	LoRA rank	Steps	τ
Qwen3.5-4B	1e−5	1e−5	32	3 / 5	0.10
Llama-3.2-3B	1e−5	1e−5	32	3 / 5	0.10
Gemma-3-4B	1e−5	3e−5	32	3 / 5	0.10

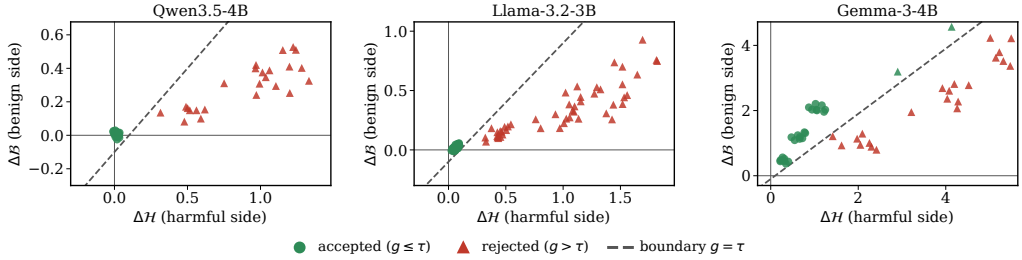


Figure 4: Reference-relative criterion geometry across models.

-0.02 to -1.2 over the probe steps, whereas Llama's are positive and already exceed τ at 3×10^{-5} (Table 4). Scaling threefold therefore keeps Gemma's benign candidates below τ, matching the 100.0 ± 0.0 (BBH-H3) and 99.89 ± 0.05 (ARC-AGI) acceptance in Table 3.

A.2 Judge Protocol

The external judge labels each response as harmful-compliant or refused and separately records lexical validity. We manually inspect a random 10% subset in each evaluation. ASR counts unsafe responses, while ASR@10 counts prompts with at least one unsafe response among ten generations.

B Gate Construction and Disclosure

The gate uses six harmful and six benign anchors. The two groups are disjoint from attack prompts and utility evaluation queries. We select short prompts that admit consistent model-specific chat formatting and evaluate a fixed compliance target by teacher forcing. Reference and candidate states use the same target, tokenizer, and formatting, so Eq. 3 isolates parameter change.

We report the construction protocol, set sizes, score, threshold, and full knowledge stress test. We withhold exact anchors and compliance targets because they constitute the deployed decision rule. The full-knowledge evaluation in Section 6.1 grants the attacker this information directly and therefore measures the strongest knowledge condition considered here.

C Cross-Model Criterion Geometry

Figure 4 plots candidates from all three model families in the $(\Delta\mathcal{H}, \Delta\mathcal{B})$ plane. Harmful updates concentrate where harmful compliance changes dominate. Benign updates remain closer to the diagonal, which permits a shared linear decision rule in the calibrated regime.