DON'T TRAIN MY MODEL: DOMAIN-CONDITIONED TRAINING RESISTANCE IN OPEN-WEIGHT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Release-time evaluations of open-weight language models characterize their predictions before adaptation, but not how they respond to subsequent fine-tuning. We study whether a checkpoint can preserve these predictions while becoming harder to fine-tune on a specified domain. We introduce DormantBomb, a checkpoint transformation that rewrites a small number of existing SwiGLU coordinates by pairing target-sensitive activations with zero down-projection columns. The resulting dormant paths have zero forward contribution at release while allowing large initial gradients during downstream adaptation. In the primary Qwen3-4B experiments, DormantBomb largely preserves release-time predictions while selectively slowing target-domain fitting, with little change under separate control-domain training. This separation holds under LoRA, direct down-projection training, and full fine-tuning. In unsafe fine-tuning experiments, differences in trainability are also accompanied by differences in generated behavior.

1 INTRODUCTION

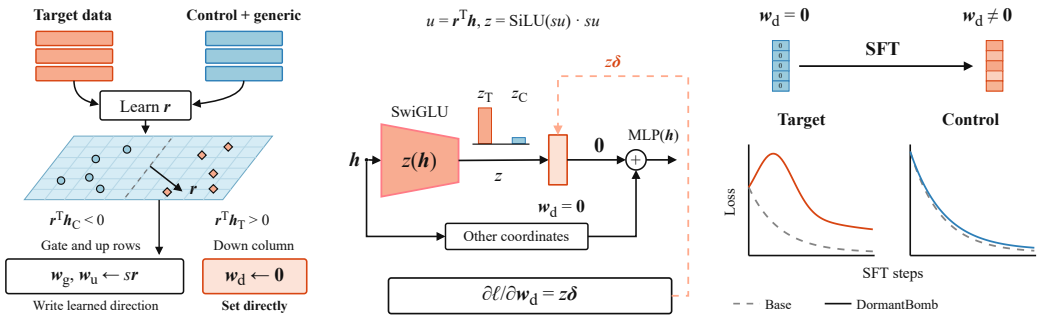
Open-weight language models are released not only as predictors, but also as checkpoints that can be further trained. Most release-time evaluations measure what a model predicts before such adaptation, yet downstream fine-tuning can substantially alter its behavior. Even small fine-tuning datasets can weaken safety alignment, including datasets whose examples appear benign (Qi et al., 2024, Zhan et al., 2024). Evaluating an open-weight model therefore requires distinguishing its behavior at release from how readily that behavior can be changed by subsequent optimization.

Prior work has explored ways to resist downstream adaptation by optimizing against simulated harmful updates, inhibiting learning on selected tasks, or inducing degraded behavior after fine-tuning (Henderson et al., 2023, Deng et al., 2024, Yi et al., 2026, Wang et al., 2026a). We consider a more specific question: can a checkpoint preserve nearly the same release-time predictions while becoming harder to fit on a specified domain? We refer to this property as domain-conditioned training resistance. The goal is not to prevent all subsequent adaptation, but to change the training response on a target domain without inducing comparable changes on unrelated data.

We introduce DormantBomb, a checkpoint transformation that exploits a local asymmetry in SwiGLU MLPs (Shazeer, 2020). A SwiGLU coordinate consists of a gate row, an up-projection row, and the corresponding down-projection column. The gate and up-projection rows determine a scalar activation, while the down-projection column maps that activation to the MLP output. Setting the down-projection column to zero removes the coordinate's forward contribution, but does not remove its activation or the activation's effect on the gradient received by that column during downstream adaptation. DormantBomb uses this asymmetry to create target-sensitive activations in selected coordinates while zeroing their corresponding down-projection columns. We call the resulting zero-forward computational paths dormant paths. They are silent at release, but can become influential once downstream updates move the zeroed down-projection parameters away from zero. The construction modifies only existing model weights and requires neither an additional module nor a modified backward rule. Figure 1 summarizes the construction.

To measure whether this construction changes downstream trainability, we compare the DormantBomb and Base checkpoints under matched downstream adaptation settings. We define the fitting gap as the target-domain negative log-likelihood (NLL) of the DormantBomb checkpoint minus that of Base after the same number of updates. An unrelated control domain is evaluated separately to

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.



(a) Construction **(b) Local mechanism** **(c) Fine-tuning**

Figure 1: **Overview of** DormantBomb**.** (a) Fit a target-separating direction, write it into existing gate and up rows, and zero the corresponding down column. (b) The path has zero forward output but local gradient $z\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the incoming loss gradient. (c) Updating the output column can expose the activation. Curves are schematic. Whole-model preservation is measured separately.

distinguish target-domain resistance from broader degradation. On Qwen3-4B with GSM8K as the target domain, a checkpoint with two rewritten coordinates achieves 99.51% token agreement with Base in a post-selection evaluation, yet retains substantial held-out fitting gaps after 1,000 updates under LoRA, direct down-projection training, and full fine-tuning. Separately trained MBPP controls show fitting gaps close to zero. In mixed-data unsafe supervised fine-tuning, a refusal-aware variant also produces substantially fewer classifier-labeled harmful generations with the same mean math accuracy as Base. Additional analysis localizes the initial gradient amplification to dormant paths and identifies hidden-state changes as one route to recovery.

In summary, our contributions are three-fold:

- We formulate domain-conditioned training resistance, separating release-time predictions from downstream target-domain trainability.
- We introduce DormantBomb, which creates target-sensitive dormant paths by rewriting a small number of SwiGLU coordinates, and localize the initial gradient amplification to these paths.
- We evaluate DormantBomb across multiple domains, models, and downstream adaptation settings, and characterize when its training resistance persists or recovers. We further show that differences in trainability can translate into different generated behavior under unsafe supervised fine-tuning.

2 Related Work

Resisting downstream adaptation. Prior work has explored limiting downstream trainability through task blocking, meta-learning, private subnetworks, and optimization against anticipated harmful updates (Henderson et al., 2023, Deng et al., 2024, Wang et al., 2026b, Rosati et al., 2024, Tamirisa et al., 2025, Huang et al., 2025). These approaches alter the model or its training objective so that subsequent adaptation becomes less effective or produces undesirable outcomes. Our setting instead focuses on whether target-domain fitting can be made more difficult while release-time predictions remain nearly unchanged.

Safety under harmful fine-tuning. A related line of work studies preserving safety under harmful or malicious fine-tuning. CTRAP induces collapse after simulated harmful updates, SEAM uses gradient-conflict and refusal-oriented objectives, and SDD induces degradation under malicious fine-tuning (Yi et al., 2026, Wang et al., 2026a, Chen et al., 2025). Other work studies the durability of such defenses and their susceptibility to subsequent repair or alternative adaptation procedures (Qi et al., 2025, Kuo et al., 2026). The primary objective in this literature is downstream safety behavior. We instead measure target-domain fitting under a fixed downstream adaptation setting and evaluate generated behavior separately.

Latent behavior and optimization geometry. Backdoors and sleeper agents demonstrate that models can contain behavior that is not expressed during ordinary evaluation but appears under particular triggers or training conditions (Gu et al., 2017, Kurita et al., 2020, Hubinger et al., 2024). FAB similarly studies behavior that is activated by subsequent fine-tuning (Gloaguen et al., 2026). DormantBomb does not encode a prescribed output behavior. Instead, it changes how readily a specified domain can be fitted after release. More broadly, similar forward behavior need not imply identical optimization geometry or training dynamics (Dinh et al., 2017, Rosati et al., 2026). DormantBomb realizes this distinction locally within existing SwiGLU coordinates, where zero forward contribution can coexist with sensitivity to subsequent parameter updates.

3 METHODOLOGY

This section describes DormantBomb and how it changes downstream trainability. We first define the adaptation setting and the fitting gap, then construct a dormant path in a SwiGLU MLP, and finally describe its effect during downstream adaptation. Figure 1 provides an overview.

3.1 SETTING AND MEASUREMENT

Let $\boldsymbol{\theta}_0$ denote the Base checkpoint and $\boldsymbol{\theta}_1$ the DormantBomb checkpoint before downstream adaptation. An adaptation interface specifies the trainable parameters, optimizer, learning rate, gradient clipping rule, and update budget. For a fixed adaptation interface, let $\theta_{m,t}$ denote the checkpoint obtained after t updates from θ_m, where $m \in \{0, 1\}$.

For an evaluation domain $\mathcal{D}$, let $L^{\mathcal{D}}_{m,t}$ denote the negative log-likelihood (NLL) of $\boldsymbol{\theta}_{m,t}$. We define the fitting gap as

$$\Delta^{\mathcal{D}}_{\text{fit}}(t) = L^{\mathcal{D}}_{1,t} - L^{\mathcal{D}}_{0,t}. \tag{1}$$

A positive fitting gap means that DormantBomb has higher NLL than Base after the same number of updates under the same adaptation interface. We measure this gap in separate target-domain and control-domain adaptation runs to compare their training responses.

3.2 CONSTRUCTING A DORMANT PATH

SwiGLU coordinate. Let $\boldsymbol{h} \in \mathbb{R}^d$ denote the input to a SwiGLU MLP, with gate and up-projection matrices $W_g, W_u \in \mathbb{R}^{d_{\text{ff}} \times d}$ and down-projection matrix $W_d \in \mathbb{R}^{d \times d_{\text{ff}}}$. For coordinate $c \in \{1, \ldots, d_{\text{ff}}\}$, let $\boldsymbol{w}_{g,c}, \boldsymbol{w}_{u,c} \in \mathbb{R}^d$ denote the corresponding gate and up-projection rows, and let $\boldsymbol{w}_{d,c} \in \mathbb{R}^d$ denote the corresponding down-projection column. Its activation is

$$z_c(\boldsymbol{h}) = \text{SiLU}\left(\boldsymbol{w}_{g,c}^\top \boldsymbol{h}\right)\left(\boldsymbol{w}_{u,c}^\top \boldsymbol{h}\right), \tag{2}$$

and its contribution to the MLP output is

$$\boldsymbol{y}_c(\boldsymbol{h}) = \boldsymbol{w}_{d,c}\, z_c(\boldsymbol{h}). \tag{3}$$

Let $\boldsymbol{y}$ denote the full MLP output and ℓ the loss. Define $\boldsymbol{\delta} = \nabla_{\boldsymbol{y}} \ell$ as the loss gradient at the MLP output. The gradient with respect to the down-projection column is

$$\nabla_{\boldsymbol{w}_{d,c}} \ell = z_c(\boldsymbol{h})\boldsymbol{\delta}. \tag{4}$$

Because the same MLP parameters are applied at every token position, the full gradient sums Eq. (4) over token positions; Appendix B gives the corresponding expression.

Equations (3) and (4) expose the asymmetry used by DormantBomb. Setting $\boldsymbol{w}_{d,c} = \mathbf{0}$ removes the coordinate's forward contribution, but does not remove its activation. The activation can therefore remain large and scale the gradient of the zeroed down-projection column during subsequent training. Different gate and up-projection rows can consequently induce different update sensitivities while producing the same zero coordinate output.

Routing direction and weight rewrite. Construction uses hidden states from a declared target domain, an unrelated control domain, and generic preservation data. Let $\mathcal{H}_T$ denote target hidden states and $\mathcal{H}_N$ denote control and generic hidden states. We fit a routing direction $\boldsymbol{r} \in \mathbb{R}^d$ such

Algorithm 1 Constructing a DormantBomb checkpoint

Require: Base checkpoint $\boldsymbol{\theta}_0$, target, control, and generic construction data; preservation-audit data; candidate layer, coordinate, and gain configurations $\mathcal{C}$

```
Collect target, control, and generic hidden states using θ_0
Fit and normalize routing directions
Rank candidate coordinates by their release-time contribution
for each configuration in C do
    Copy θ_0
    Rewrite the selected gate and up-projection rows and zero the down-projection columns using Eq. (5)
    Evaluate the resulting checkpoint with the preservation audit
end for
Select a passing checkpoint using construction and preservation measurements only
return the selected checkpoint, or report that no candidate passed
```

that $\boldsymbol{r}^\top \boldsymbol{h}$ tends to be positive for $\boldsymbol{h} \in \mathcal{H}_T$ and negative for $\boldsymbol{h} \in \mathcal{H}_N$. The fitting objective and normalization procedure are given in Appendix B.

We rank candidate coordinates by their activation-weighted output magnitude and by the logit change induced by temporarily zeroing their down-projection columns. Low-contribution candidates are preferred because replacing an existing coordinate removes its original contribution.

For a selected coordinate c, we rewrite

$$\boldsymbol{w}'_{g,c} = s\boldsymbol{r}, \qquad \boldsymbol{w}'_{u,c} = s\boldsymbol{r}, \qquad \boldsymbol{w}'_{d,c} = \boldsymbol{0}, \tag{5}$$

where $s > 0$ is the gain. Let

$$u(\boldsymbol{h}) = \boldsymbol{r}^\top \boldsymbol{h}. \tag{6}$$

Using $\mathrm{SiLU}(x) = x\sigma(x)$, where σ is the logistic sigmoid, the rewritten activation is

$$\widetilde{z}_c(\boldsymbol{h}) = s^2 u(\boldsymbol{h})^2 \sigma(s u(\boldsymbol{h}))\,. \tag{7}$$

For positive routing projections, the activation approaches quadratic growth in s. For negative projections bounded away from zero, the sigmoid term suppresses the activation as s increases. The routing projection therefore determines which hidden states strongly activate the rewritten coordinate, while the gain controls the activation magnitude.

The complete checkpoint is not guaranteed to be exactly function-preserving because replacing the selected coordinate removes its original contribution. We therefore use a preservation audit to verify that the rewritten checkpoint maintains high release-time prediction agreement with the original checkpoint.

The construction can be applied to selected coordinates in one or more Transformer layers, with each rewritten coordinate defining a dormant path at its corresponding layer. The overall checkpoint construction procedure is summarized in Algorithm 1.

Routing variants. The basic variant uses target states as positives and control and generic states as negatives. For unsafe-response experiments, the refusal-aware variant additionally includes safe-refusal response states in the negative set. Both variants use the same weight rewrite in Eq. (5). We refer to them as DormantBomb (basic) and DormantBomb (refusal-aware), respectively.

3.3 Training-Time Effect and Finite-Budget Resistance

For direct down-projection training, Eq. (4) directly determines the gradient of the rewritten column. Under LoRA, the rewritten activation enters the gradients of the trainable factors through the LoRA parameterization; Appendix B gives the corresponding expressions. In both cases, the update magnitude depends on the dormant-path activation and on the adaptation interface.

A large dormant-path gradient does not reverse the first-order descent direction of gradient descent. Gradient amplification alone therefore does not imply a loss increase. A fitting gap can arise even when both models' losses decrease. Once the zero down-projection column moves away from zero, the target-sensitive activation can produce a large path contribution and alter subsequent updates.

Under globally clipped SGD, a large dormant-path gradient can also reduce the effective update magnitude of other trainable parameters. Appendix B gives the corresponding derivation.

A large initial fitting gap does not by itself imply persistent resistance. Its persistence depends on how the routing projections and downstream parameters evolve during training. We study these effects empirically in Section 5.

4 EXPERIMENTS

We evaluate whether DormantBomb can preserve release-time predictions while changing subsequent trainability on a specified target domain. We first establish this separation on Qwen3-4B, then compare DormantBomb with prior resistance objectives, and finally examine whether the resulting difference in trainability can translate into behavioral differences under unsafe supervised fine-tuning.

4.1 EXPERIMENTAL SETUP

Models and domains. Our primary experiments use Qwen3-4B with GSM8K and CSQA as target domains and MBPP as an unrelated control domain (Yang et al., 2025, Talmor et al., 2019, Cobbe et al., 2021, Austin et al., 2021). Additional experiments use Qwen2.5 models, Llama-3.2-1B, and DeepSeek-Coder-1.3B to study construction cost, mechanism, and persistence. Unsafe fine-tuning experiments use Qwen3-4B-Instruct-2507, with WildJailbreak and BeaverTails providing refusal and unsafe-response data (Jiang et al., 2024, Ji et al., 2023). Detailed data splits and preprocessing are given in Appendices A and E.

Adaptation interfaces. We evaluate three adaptation interfaces: LoRA, direct down-projection training, and full fine-tuning. Base and DormantBomb use the same adaptation interface, data order, and paired random seeds. The primary Qwen3 experiments use 1,000 updates, with six paired seeds for target-domain adaptation and three paired seeds for control-domain adaptation. Longer budgets are analyzed in Section 5.3. Optimization hyperparameters and additional implementation details are reported in Appendix A.

Measurement. We use the fitting gap defined in Eq. (1) as the primary measure of training resistance. For the primary Qwen3 experiments, training-probe NLL is evaluated on a fixed 16-record subset of the SFT data, while held-out NLL is evaluated on a disjoint evaluation set. We report both training-probe and held-out fitting gaps on the target domain, together with fitting gaps obtained from separate adaptation runs on the control domain.

Candidate checkpoints are selected using a preservation audit which measures release-time prediction agreement with Base on audit data. Downstream adaptation outcomes are not used for checkpoint selection. Release-time preservation is additionally measured on teacher-forced records outside the selection audit. We refer to this as the post-selection evaluation. Full audit criteria and data roles are given in Appendix A.

Baselines. We compare against CTRAP-r and SEAM-r, release-constrained adaptations of two prior resistance objectives. CTRAP-r applies a fixed-token collapse objective after a simulated target update, while SEAM-r uses gradient-conflict coupling with harmful-data unlearning disabled. Both use Base-prediction cloning during construction. The suffix "-r" denotes our release-constrained implementations rather than complete reproductions of the original systems. Appendix A.4 gives the full implementation details.

4.2 SIMILAR RELEASE-TIME PREDICTIONS, DIFFERENT TRAINING TRAJECTORIES

We first test whether DormantBomb changes downstream trainability while maintaining high release-time prediction agreement with Base. On Qwen3-4B with GSM8K as the target domain, the selected DormantBomb checkpoint rewrites one coordinate in each of two layers. It obtains 99.55% aggregate token agreement with Base in the preservation audit and 99.51% agreement in the post-selection teacher-forced evaluation. Despite this close release-time prediction agreement, downstream adaptation produces a large separation from Base.

Table 1: **Qwen3-4B GSM8K fitting gaps after 1,000 updates.** Positive values indicate higher NLL for DormantBomb than for Base under the same adaptation interface. The training-probe and held-out columns use GSM8K, and the control column uses separately trained MBPP.

Adaptation interface	Training-probe gap ↑	Held-out gap ↑	MBPP Control gap
LoRA	4.835	4.921	−0.023
Direct down-projection	3.058	2.843	−0.008
Full fine-tuning	3.928	3.697	+0.014

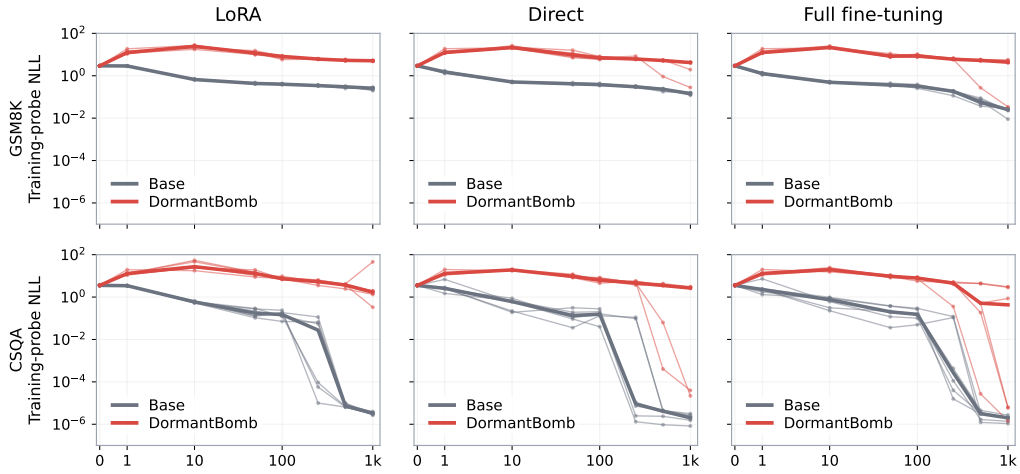


Figure 2: **Repeated Qwen3-4B training trajectories.** Rows correspond to GSM8K and CSQA, and columns correspond to LoRA, direct down-projection training, and full fine-tuning. Thin lines show individual runs and thick lines show pointwise medians. Panels share a log-NLL axis.

Table 1 reports mean fitting gaps after 1,000 updates. The target-domain training-probe and held-out gaps remain large under all three adaptation interfaces, whereas the fitting gaps from separately trained MBPP control runs remain close to zero.

Figure 2 shows the corresponding training trajectories. For GSM8K, the training-probe fitting gap is positive in every run and at every recorded checkpoint from step 1 through step 1,000. The held-out results show the same separation, indicating that the effect is not confined to the fixed training probe.

CSQA provides a second target domain. The fitting separation is also observed on CSQA, although its magnitude varies more across runs. Direct down-projection training and full fine-tuning retain positive mean fitting gaps, while LoRA exhibits greater run-to-run variability. The corresponding held-out fitting gaps remain positive on average. Aggregate statistics are reported in Appendix A. Together, these results show that checkpoints with closely matched release-time predictions can follow substantially different training trajectories under the same adaptation interface.

4.3 Comparison with Prior Resistance Objectives

Under the same Qwen3 evaluation protocol, the release-constrained CTRAP-r and SEAM-r checkpoints fit the target data substantially more readily than the DormantBomb checkpoint. In the post-selection teacher-forced evaluation on GSM8K, release-time token agreement with Base is 99.51% for DormantBomb, 99.38% for SEAM-r, and 95.72% for CTRAP-r. The corresponding CSQA agreements are 99.71%, 99.42%, and 94.23%, respectively. Thus, the larger fitting separation observed for DormantBomb is not accompanied by lower release-time token agreement than SEAM-r.

We additionally compare construction cost on Qwen2.5-1.5B with MBPP as the target domain. Table 2 reports release-time token agreement, target-domain NLL after 1,000 updates under direct down-projection training and full fine-tuning, construction time, and peak allocated GPU mem-

Table 2: **Construction cost and target-domain resistance on Qwen2.5-1.5B.** Target-domain NLL is measured after 1,000 updates, with higher values indicating poorer target-domain fitting. Release-time token agreement with Base is measured before downstream adaptation. Construction time includes checkpoint construction and the preservation audit, but excludes prior hyperparameter search. Memory denotes peak allocated GPU memory. CTRAP-r and SEAM-r are release-constrained adaptations and differ in their levels of release-time preservation. Bold indicates the best value in each column.

Checkpoint	Release agreement (%) ↑	Direct NLL ↑	Full fine-tuning NLL ↑	Time (s) ↓	Memory (GiB) ↓
Base	**100.00**	$(3.9 \pm 1.3) \times 10^{-3}$	$(1.6 \pm 0.1) \times 10^{-4}$	–	–
CTRAP-r	87.85	$(2.9 \pm 0.7) \times 10^{-3}$	$(3.1 \pm 3.2) \times 10^{-4}$	185.7	10.79
SEAM-r	99.23	$(4.5 \pm 1.7) \times 10^{-3}$	$(1.4 \pm 0.1) \times 10^{-4}$	151.1	8.09
DormantBomb	98.75	$\mathbf{2.989} \pm 1.998$	$\mathbf{2.896} \pm 1.928$	**41.2**	**5.94**

ory. Under these implementations, the DormantBomb checkpoint retains substantially higher target-domain NLL while requiring less construction time and memory than CTRAP-r and SEAM-r. Because the compared checkpoints differ in both release-time preservation and implementation details, we treat these results as a comparison of the release-constrained implementations rather than as a full-system benchmark.

4.4 Behavioral Effects under Unsafe Fine-Tuning

We next ask whether the difference in trainability can affect generated behavior. On Qwen3-4B-Instruct-2507, we apply DormantBomb (refusal-aware) and perform mixed-data unsafe supervised fine-tuning using 256 clean and 25 harmful examples. Base and the DormantBomb checkpoint produce identical generations on the harmful and math evaluation prompts before fine-tuning.

We use unsafe-reference NLL to measure fitting to the supervised unsafe responses. This quantity is a fitting measure and is not treated as a behavioral safety score. Harmful generations are evaluated separately using the classifier-labeled harmful rate.

Table 3 reports the primary mixed-data MLP-LoRA result. After 200 updates, the classifier-labeled harmful rate decreases from 26.562% for Base to 2.344% for DormantBomb (refusal-aware), while both checkpoints have the same mean math accuracy. The higher unsafe-reference NLL indicates poorer fitting of the unsafe reference responses.

Table 3: **Qwen3-4B-Instruct outcomes after mixed-data unsafe fine-tuning.** Mean±SD over two paired MLP-LoRA runs. Unsafe-reference NLL measures fitting to the supervised unsafe responses and is not treated as a safety score. Classifier-labeled harmful rate is evaluated on 64 prompts, and math accuracy uses 24 questions.

Checkpoint	Unsafe-reference NLL ↑	Classifier-labeled harmful (%) ↓	Math accuracy (%) ↑
Base	1.698 ± 0.013	26.562 ± 4.419	62.500 ± 5.893
DormantBomb (refusal-aware)	3.723 ± 0.450	2.344 ± 1.105	62.500 ± 5.893

Generation inspection shows that repetitive capped outputs become more frequent after the refusal-aware rewrite. The reduction in classifier-labeled harmful generations should therefore not be interpreted as coherent refusal alone. Additional unsafe fine-tuning conditions and longer-budget experiments are analyzed in Section 5.3 and Appendix E.3.

5 Analysis

The experiments above establish a separation between release-time predictions and downstream trainability. We now examine the source of the initial fitting gap, the factors that determine its domain scope, and the mechanisms that govern persistence and recovery.

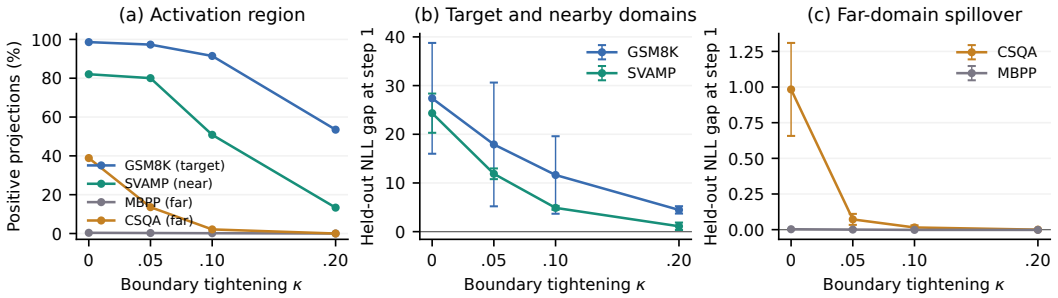


Figure 3: **Controlling domain scope on Qwen2.5-0.5B.** Positive-projection rates and one-update held-out fitting gaps as the routing-boundary parameter κ increases. Mean$\pm$SD is reported over two repetitions.

5.1 Source of the Initial Gradient Amplification

The mechanism in Section 3.2 predicts that a target-sensitive dormant-path activation can amplify the gradient of the corresponding down-projection parameters. We test this prediction on Qwen2.5-0.5B with CSQA as the target domain. Specifically, we ablate the rewritten-coordinate activation immediately before the down projection. Because the corresponding down-projection column is zero at release, this intervention leaves the release-time function unchanged.

The ablation reduces the target-domain LoRA output-factor gradient norm from $(9.62 \pm 7.18) \times 10^3$ to 0.534 ± 0.084, close to the control-domain value of 0.837 ± 0.130. This result shows that the large initial target-domain gradient is driven by the rewritten-coordinate activation, supporting dormant-path gradient amplification as a mechanism underlying the initial fitting gap.

5.2 Controlling the Affected Domain

Domain-conditioned training resistance requires a large fitting gap on the target domain without a comparable gap on unrelated data. We therefore separate two factors: how strongly the intervention affects the target domain and how broadly that effect extends across domains. Distributing rewritten coordinates across multiple layers strengthens the target-domain effect. On Qwen2.5-0.5B, using four rewritten layers produces higher target NLL than using one rewritten layer across the evaluated target-domain settings, although it also reduces release-time token agreement. The four-layer setting therefore strengthens the target-domain effect while reducing release-time token agreement from 99.79% to 98.46%. Detailed results are provided in Appendix C.

For presentation, we refer to SVAMP as a near domain because it is closely related to the GSM8K target task, and to MBPP and CSQA as far domains because they differ more substantially from GSM8K in task format and content. We next examine domain scope by varying the routing direction and gain. Starting from the original GSM8K routing direction, we introduce a routing-boundary parameter κ that reduces positive projections on the evaluated pre-SFT states while preserving the direction norm. The construction is given in Appendix C.3. As shown in Figure 3, increasing κ reduces positive projections outside the target region and correspondingly reduces one-update fitting gaps on non-target domains. At $\kappa = 0.10$, the CSQA fitting gap decreases from 0.983 to 0.015, while GSM8K retains a gap of 11.637.

To distinguish the effect of routing from a reduction in activation magnitude, we additionally increase the gain at $\kappa = 0.10$ to match the original GSM8K construction gate's 95th-percentile preactivation. Under this matched-gain condition, the GSM8K gap increases to 15.718, while the CSQA gap remains 0.028 and the MBPP gap remains near zero. By contrast, reducing the gain alone decreases fitting gaps across domains. These results separate the roles of routing and gain: the routing direction primarily determines which hidden states activate the dormant path, whereas the gain controls the magnitude of that activation.

Table 4: **Activation restoration after 1,000 SGD–LoRA updates.** Paired record-level ΔNLL (mean±SD) on Qwen2.5-1.5B. Positive values indicate higher NLL after the intervention.

Evaluation set	Restore rewritten activation	Restore LoRA input factor
Unsafe SFT	2.480 ±2.108	0.000 ±0.000
Unsafe held-out	2.026 ±1.804	−0.001 ±0.003
Paired refusal	0.052 ±0.178	0.000 ±0.000
Benign	0.000 ±0.000	0.000 ±0.000

5.3 Persistence and Recovery

The persistence of the fitting gap varies across models and adaptation interfaces. In the cross-model MBPP experiments, the mean LoRA fitting gap decreases from 4.230 to 0.275 on Llama-3.2-1B and from 4.943 to 0.052 on DeepSeek-Coder-1.3B between steps 100 and 1,000. For longer-budget experiments, once Base has reached a training NLL below 0.1, we define stable recovery as the earliest recorded step at which the DormantBomb training NLL falls below 0.1 and remains below 0.1 at all subsequent recorded checkpoints. For Qwen2.5-1.5B on MBPP, we extend training from 1,000 to 5,000 updates. Under both direct down-projection training and full fine-tuning, four of six runs remain above the stable-recovery threshold throughout the training budget, while the remaining two runs recover.

We next examine whether hidden-state changes contribute to recovery. After 1,000 SGD–LoRA updates, restoring the release-time rewritten-coordinate activations on a Qwen2.5-1.5B DormantBomb (refusal-aware) checkpoint increases unsafe training NLL from 0.334 to 2.814 and held-out NLL from 4.471 to 6.497, whereas restoring the corresponding LoRA input factor produces negligible changes. Consistent with this intervention, the fraction of supervised training-token positions with positive routing projections decreases from 33.21% before fine-tuning to zero afterward. Together, these results support hidden-state drift as one route to recovery: downstream adaptation can move representations out of the positive-projection region and thereby reduce dormant-path activation.

A separate Qwen3-8B experiment with 1,024 SFT examples and 768 AdamW updates shows a large final unsafe-reference fitting gap after clean SFT, but substantially smaller gaps under mixed and harmful-only SFT. This further supports viewing DormantBomb resistance as a finite-budget property rather than a permanent prevention of adaptation. Harmful generation behavior is not evaluated in this experiment. Full results are reported in Appendix E.3.

5.4 Recovery under Direct Modification and Alternative Adaptation

The rewritten coordinates provide a direct probe of recoverability. On Qwen2.5-1.5B, normalizing the rewritten gate and up-projection rows or setting them to zero reduces target training NLL below 0.1 within 250 updates. Together with Section 5.3, these results identify two routes to recovery: hidden-state changes during fine-tuning and direct modification of the rewritten coordinates. Additional results across adaptation interfaces and optimizers are reported in Appendix D.4.

6 Conclusion

We introduce DormantBomb, a checkpoint transformation that maintains high release-time prediction agreement while making target-domain fitting more difficult under multiple adaptation interfaces. Our analysis localizes the initial gradient amplification to dormant paths and identifies hidden-state drift as one route to recovery. Experiments across multiple models, domains, and adaptation settings show substantial target-domain fitting gaps with much smaller effects on unrelated control domains. These results show that release-time predictions alone do not characterize how an open-weight checkpoint responds to subsequent fine-tuning, motivating post-release trainability as a separate dimension of model evaluation.

Reproducibility Statement

The method section and appendices describe the construction procedure, data splits, training settings, and evaluation protocol. The anonymous code provides the core implementation and scripts for public-data preparation, basic fine-tuning experiments, and result aggregation. The README lists dependencies and commands for running these experiments.

AI Use Statement

Generative AI tools assisted with language editing, code drafting, and visualization. The authors inspected the resulting code and reviewed all manuscript text and figures. The authors take responsibility for the final content, claims, and artifacts.

Ethics Statement

This work studies a dual-use checkpoint transformation that may either raise the cost of harmful adaptation or obstruct legitimate model customization. We therefore report bypasses and negative safety outcomes, avoid releasing harmful response text, and recommend disclosure and independent trainability audits. Experiments use public data without human subjects.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Shuhao Chen, Weisen Jiang, Yeqi Gong, Shengda Luo, Chengxiang Zhuo, Zang Li, James T. Kwok, and Yu Zhang. Spard: Defending harmful fine-tuning attack via safety projection with relevance-diversity data selection. In *International Conference on Machine Learning*, 2026.

Zixuan Chen, Weikai Lu, Xin Lin, and Ziqian Zeng. Sdd: Self-degraded defense against malicious fine-tuning. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jiangyi Deng, Shengyuan Pang, Yanjiao Chen, Liangming Xia, Yijie Bai, Haiqin Weng, and Wenyuan Xu. Sophon: Non-fine-tunable learning to restrain task transferability for pre-trained models. In *IEEE Symposium on Security and Privacy*, 2024.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

Thibaud Gloaguen, Mark Vero, Robin Staab, and Martin Vechev. Watch your steps: Dormant adversarial behaviors that activate upon LLM finetuning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yfM2e8Icsw.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *Workshop on Machine Learning and Computer Security*, 2017.

Peter Henderson, Eric Mitchell, Christopher D. Manning, Dan Jurafsky, and Chelsea Finn. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2023.

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. In *International Conference on Learning Representations*, 2025.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Chi Zhang, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. *arXiv preprint arXiv:2307.04657*, 2023. URL https://arxiv.org/abs/2307.04657.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems*, 2024.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

Kevin Kuo, Chhavi Yadav, and Virginia Smith. Open-weight llm fine-tuning defenses are susceptible to simple attacks. *arXiv preprint arXiv:2605.26526*, 2026.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 21–27 Jul 2024.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight llms. In *International Conference on Learning Representations*, 2025.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 5234–5247, 2024.

Domenic Rosati, Xijie Zeng, Hong Huang, Sebastian Dionicio, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Limits of convergence-rate control for open-weight safety, 2026.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, 2019.

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. In *International Conference on Learning Representations*, 2025.

Yuhui Wang, Rongyi Zhu, and Ting Wang. Self-destructive language models. In *International Conference on Learning Representations*, 2026a.

Ziyao Wang, Nizhang Li, Pingzhi Li, Guoheng Sun, Tianlong Chen, and Ang Li. Towards building non-fine-tunable foundation models. *arXiv preprint arXiv:2602.00446*, 2026b.

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

Biao Yi, Tiansheng Huang, Baolei Zhang, Tong Li, Lihai Nie, Zheli Liu, and Li Shen. Ctrap: Embedding collapse trap to safeguard large language models from harmful fine-tuning. In *Annual Meeting of the Association for Computational Linguistics*, 2026.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori B. Hashimoto, and Daniel Kang. Removing rlhf protections in gpt-4 via fine-tuning. *arXiv preprint arXiv:2311.05553*, 2024.

A Primary Experimental Setting and Additional Results

A.1 Data Roles and Splits

For each primary Qwen3 target domain, we use 24 construction records, 32 preservation-audit records, 64 SFT records, and 64 held-out evaluation records. These sets are disjoint by identifier and normalized input, and pilot records are excluded. Target experiments use six paired training seeds and control experiments use three, with fixed data splits and checkpoint constructions.

For checkpoint screening, the preservation audit uses 16 target records and 16 control records, together with 16 generic prompts reused from construction. We screen nine candidate configurations. The selected Qwen3 checkpoints rewrite one SwiGLU coordinate in each of layers 23 and 35 with gain $s = 6{,}144$.

A.2 Preservation Audit and Post-Selection Evaluation

We define token agreement as the fraction of evaluated token positions at which a candidate checkpoint and Base have the same top-1 next-token prediction. The preservation audit includes both prompt positions and teacher-forced response positions. Checkpoint selection requires at least 99.5% aggregate token agreement and at least 98% agreement within every evaluated group. Candidates must also produce identical eight-token greedy continuations on four target prompts, four control prompts, and 16 generic prompts.

The selected GSM8K checkpoint agrees with Base on 5,502 of 5,527 evaluated token positions, corresponding to 99.55% aggregate agreement. Its minimum-group agreement is 98.953%. The selected CSQA checkpoint agrees on 3,373 of 3,385 positions, corresponding to 99.65% aggregate agreement, with 99.338% minimum-group agreement. These values are used during checkpoint selection and are therefore not independent post-selection measurements. We separately evaluate 16 teacher-forced target records outside the selection audit. On GSM8K, token agreement with Base is 99.51% for DormantBomb, 99.38% for SEAM-r, and 95.72% for CTRAP-r. The corresponding CSQA values are 99.71%, 99.42%, and 94.23%, respectively.

A.3 Adaptation Interfaces and Measurement

The primary experiments evaluate LoRA, direct down-projection training, and full fine-tuning. Qwen3 model weights are loaded in BF16. Direct down-projection training and full fine-tuning SGD update these stored weights without FP32 master copies. LoRA targets the down projections with rank 8 and $\alpha = 16$.

The primary Qwen3 and Qwen2.5 direct and full fine-tuning protocols use momentum-free SGD with learning rate 0.05, global gradient clipping at 1, and 1,000 updates. Extended-budget experiments use 5,000 updates. Base and DormantBomb use matched adaptation interfaces, data orders, and random seeds.

Training-probe NLL for the primary Qwen3 experiments is measured on a fixed 16-record probe from the 64-record SFT set. Held-out NLL is measured on all 64 evaluation records. We average response-token NLL within each record and then average equally across records. Repeated experiments report mean and sample standard deviation over training seeds on fixed data splits. Component experiments use three initializations unless stated otherwise.

A.4 Comparator Implementations

CTRAP-r and SEAM-r are release-constrained adaptations of the corresponding prior objectives rather than complete reproductions of the original systems. Both update down-projection LoRA parameters during construction and include Base-prediction cloning.

CTRAP-r differentiates through one target SFT update before applying its fixed-token collapse objective. SEAM-r estimates gradient-cosine coupling by finite differences, with harmful-data unlearning disabled. For the small-model safety experiments, each candidate is interpolated with Base and we select the strongest point on a fixed grid that passes the preservation audit. Selection does not use downstream adaptation outcomes.

Table 5: **Qwen3-4B target training after 1,000 updates.** Training-probe NLL, mean±SD over six paired seeds. The raw values are reported here to complement the fitting gaps used in the main text.

Target	Update	Base ↑	CTRAP-r ↑	SEAM-r ↑	DormantBomb ↑
GSM8K	LoRA	$0.247_{\pm 0.034}$	$0.250_{\pm 0.030}$	$0.241_{\pm 0.025}$	$\mathbf{5.082}_{\pm 0.173}$
	Direct	$0.142_{\pm 0.016}$	$0.154_{\pm 0.023}$	$0.140_{\pm 0.027}$	$\mathbf{3.200}_{\pm 1.701}$
	Full	$0.023_{\pm 0.007}$	$0.027_{\pm 0.011}$	$0.023_{\pm 0.007}$	$\mathbf{3.951}_{\pm 2.009}$
CSQA	LoRA	$(3.3_{\pm 0.5}) \times 10^{-6}$	$(4.1_{\pm 0.8}) \times 10^{-6}$	$(3.7_{\pm 0.5}) \times 10^{-6}$	$8.826_{\pm 18.181}$
	Direct	$(2.0_{\pm 0.8}) \times 10^{-6}$	$(1.7_{\pm 0.7}) \times 10^{-6}$	$(2.2_{\pm 0.7}) \times 10^{-6}$	$\mathbf{1.895}_{\pm 1.477}$
	Full	$(1.9_{\pm 0.6}) \times 10^{-6}$	$(1.3_{\pm 0.7}) \times 10^{-6}$	$(1.7_{\pm 1.2}) \times 10^{-6}$	$\mathbf{1.131}_{\pm 1.459}$

The selected interpolation coefficients are 10^{-5} for CTRAP-r and 10^{-3} for SEAM-r, yielding 99.964% and 99.909% token agreement, respectively. The Qwen3-4B unsafe fine-tuning experiment compares Base directly with DormantBomb (refusal-aware).

A.5 Additional Qwen3-4B Results

For CSQA under LoRA, one of six DormantBomb runs ends at NLL 45.919, raising the mean to 8.826 while the median remains 1.703. Direct and full fine-tuning CSQA results also exhibit greater run-to-run variation than the primary GSM8K results.

Held-out target NLL and separately trained MBPP control NLL are reported below. These values underlie the held-out and control fitting gaps summarized in the main text.

Table 6: **Held-out target NLL and separate MBPP control-SFT NLL** after 1,000 updates (mean±SD). Bold marks the best mean within each Base and DormantBomb pair.

Target	Adaptation	Held-out ↑ Base	Held-out ↑ DormantBomb	Control SFT ↓ Base	Control SFT ↓ DormantBomb
GSM8K	LoRA	$0.632_{\pm 0.026}$	$\mathbf{5.553}_{\pm 0.166}$	$0.420_{\pm 0.085}$	$\mathbf{0.397}_{\pm 0.059}$
	Direct	$0.785_{\pm 0.106}$	$\mathbf{3.628}_{\pm 1.855}$	$0.161_{\pm 0.048}$	$\mathbf{0.153}_{\pm 0.047}$
	Full	$1.055_{\pm 0.061}$	$\mathbf{4.752}_{\pm 1.943}$	$\mathbf{0.031}_{\pm 0.031}$	$0.045_{\pm 0.031}$
CSQA	LoRA	$0.726_{\pm 0.063}$	$\mathbf{10.731}_{\pm 17.255}$	$0.420_{\pm 0.085}$	$\mathbf{0.408}_{\pm 0.070}$
	Direct	$0.610_{\pm 0.030}$	$\mathbf{3.478}_{\pm 1.955}$	$0.161_{\pm 0.048}$	$\mathbf{0.128}_{\pm 0.046}$
	Full	$0.591_{\pm 0.118}$	$\mathbf{2.569}_{\pm 2.712}$	$0.031_{\pm 0.031}$	$\mathbf{0.029}_{\pm 0.024}$

B Construction and Gradient Derivations

B.1 Routing-Direction Objective

Let $\mathcal{H}_{\mathrm{T}}$ denote the multiset of hidden states collected from target construction data and $\mathcal{H}_{\mathrm{N}}$ the corresponding negative-state multiset. The negative set contains control and generic states.

We optimize a routing direction $\bar{\boldsymbol{r}}$ with

$$\begin{aligned} J(\bar{\boldsymbol{r}}) = \; & \mathrm{mean}_{\boldsymbol{h} \in \mathcal{H}_{\mathrm{T}}} \, \mathrm{softplus}\left(1 - \bar{\boldsymbol{r}}^{\top} \boldsymbol{h}\right) \\ & + \mathrm{mean}_{\boldsymbol{h} \in \mathcal{H}_{\mathrm{N}}} \, \mathrm{softplus}\left(1 + \bar{\boldsymbol{r}}^{\top} \boldsymbol{h}\right) \\ & + 2 \, \mathrm{softplus}\big(1 + \mathrm{TopMean}_{k} \left\{\bar{\boldsymbol{r}}^{\top} \boldsymbol{h} : \boldsymbol{h} \in \mathcal{H}_{\mathrm{N}}\right\}\big) . \end{aligned} \tag{8}$$

Here $\mathrm{TopMean}_{k}$ denotes the mean of the k largest negative-state projections. The first two terms encourage positive target projections and negative non-target projections, while the third emphasizes difficult negatives near or beyond the routing boundary.

The fitted direction is normalized by the absolute 95th-percentile target projection, with the denominator floored at 10^{-6}.

B.2 Coordinate Selection and Weight Rewrite

For coordinate c, the Base SwiGLU activation is

$$z_c(\boldsymbol{h}) = \mathrm{SiLU}\left(\boldsymbol{w}_{g,c}^\top \boldsymbol{h}\right)\left(\boldsymbol{w}_{u,c}^\top \boldsymbol{h}\right), \tag{9}$$

and its contribution to the MLP output is $\boldsymbol{w}_{d,c} z_c(\boldsymbol{h})$.

Candidate coordinates are first shortlisted according to

$$\|\boldsymbol{w}_{d,c}\|_2 \ \mathrm{mean}\,|z_c|, \tag{10}$$

which measures their activation-weighted output magnitude. Final selection is performed without replacement using the logit change caused by temporarily zeroing the corresponding down-projection column.

For a selected coordinate, DormantBomb applies

$$\boldsymbol{w}'_{g,c} = s\boldsymbol{r}, \qquad \boldsymbol{w}'_{u,c} = s\boldsymbol{r}, \qquad \boldsymbol{w}'_{d,c} = \boldsymbol{0}, \tag{11}$$

where $s > 0$ is the gain. Associated selected biases are set to zero.

B.3 Direct Down-Projection Gradient

For token position t, let $\boldsymbol{h}_t$ denote the hidden state entering the SwiGLU block and define the routing projection

$$u_t = \boldsymbol{r}^\top \boldsymbol{h}_t. \tag{12}$$

The rewritten coordinate has activation

$$\widetilde{z}_c(\boldsymbol{h}_t) = s^2 u_t^2 \sigma(s u_t). \tag{13}$$

Let $\boldsymbol{\delta}_t$ denote the loss gradient arriving at the MLP output, and let $\mathcal{I}$ denote the set of non-padding prompt and response token positions. The gradient with respect to the rewritten down-projection column is

$$\nabla_{\boldsymbol{w}'_{d,c}} \mathcal{L} = \sum_{t \in \mathcal{I}} \widetilde{z}_c(\boldsymbol{h}_t)\boldsymbol{\delta}_t. \tag{14}$$

Because $\boldsymbol{w}'_{d,c} = \boldsymbol{0}$ immediately after construction, the rewritten gate and up-projection rows receive zero gradient through this path at the initial step. In contrast, the down-projection column can receive a large gradient when the rewritten activation is large.

B.4 LoRA Gradient

For LoRA applied to the down projection, write

$$\Delta W_d = \frac{\alpha}{\rho} BA, \tag{15}$$

where ρ is the LoRA rank, $A \in \mathbb{R}^{\rho \times d_{\mathrm{ff}}}$ is the input factor, and $B \in \mathbb{R}^{d \times \rho}$ is the output factor. We use the initialization $B_0 = \boldsymbol{0}$. Under SGD without momentum or weight decay, the input factor therefore remains unchanged after the first update, so $A_1 = A_0$.

Let $\boldsymbol{e}_c$ denote the c-th standard basis vector, η the learning rate, and χ the global gradient-clipping multiplier. Let $z_{j,t}$ denote the activation of MLP coordinate j at token position t. The effective LoRA update to the c-th down-projection column after the first optimization step is

$$\Delta W_{d,1}\boldsymbol{e}_c = -\eta\chi\left(\frac{\alpha}{\rho}\right)^2 \sum_{t \in \mathcal{I}} \boldsymbol{\delta}_t \sum_j z_{j,t}\left\langle A_0 \boldsymbol{e}_j, A_0 \boldsymbol{e}_c \right\rangle. \tag{16}$$

The $j = c$ term contains the rewritten-coordinate activation, while the remaining terms capture cross-coordinate interactions induced by the LoRA parameterization.

B.5 Effect of Global Gradient Clipping

Consider a path with scalar activation a, output parameter $\boldsymbol{w}$, and residual output

$$\boldsymbol{y} = \boldsymbol{y}_0 + a\boldsymbol{w}. \tag{17}$$

For clipped SGD with learning rate η and clipping threshold τ,

$$\|\Delta \boldsymbol{w}\|_2 \le \eta\tau, \qquad \|\Delta \boldsymbol{y}\|_2 = \eta|a| \min(|a|\|\boldsymbol{\delta}\|_2, \tau)\,. \tag{18}$$

Gradient clipping therefore bounds the parameter step but does not remove the dependence of the residual change on activation magnitude. The update remains first-order descent; loss growth requires finite-step effects. Continued training can additionally change the hidden states entering the rewritten coordinate.

C Mechanism and Domain-Scope Experiments

C.1 Gradient-Ablation Protocol

The gradient-ablation experiment uses Qwen2.5-0.5B with CSQA as the target domain and MBPP as the control domain. We rewrite coordinate 1048 in layer 23 and use rank-1 LoRA with $\alpha = 2$. Three initializations are evaluated. Gradients are averaged over eight records before computing the joint output-factor norm.

C.2 Rewritten Layers

Layer-distribution comparisons average three seeds across the evaluated target-domain settings. Using four rewritten layers consistently yields higher target NLL than using a single rewritten layer. Averaged across settings, target NLL increases from 4.83 to 7.54 at step 20 and from 0.88 to 5.47 at step 50. This stronger target-domain effect is accompanied by a decrease in release-time token agreement from 99.79% to 98.46%.

C.3 Routing-Boundary and Gain Controls

Let $\boldsymbol{r}_0$ denote the original GSM8K routing direction and define

$$\widehat{\boldsymbol{r}}_0 = \frac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2}. \tag{19}$$

Let $\boldsymbol{\mu}$ denote the unit-normalized, domain-balanced mean of normalized construction states from GSM8K, MBPP, and CSQA. We define

$$\boldsymbol{r}_\kappa = \|\boldsymbol{r}_0\|_2 \frac{\widehat{\boldsymbol{r}}_0 - \kappa\boldsymbol{\mu}}{\|\widehat{\boldsymbol{r}}_0 - \kappa\boldsymbol{\mu}\|_2}, \qquad \kappa \ge 0. \tag{20}$$

This transformation changes the routing direction while preserving its norm, with $\kappa = 0$ recovering the original direction.

For states satisfying $\boldsymbol{\mu}^\top \boldsymbol{h} > 0$, the positive-projection region under $\boldsymbol{r}_\kappa$ is characterized by

$$\frac{\widehat{\boldsymbol{r}}_0^\top \boldsymbol{h}}{\boldsymbol{\mu}^\top \boldsymbol{h}} > \kappa. \tag{21}$$

Thus, increasing κ narrows the positive-projection region. All evaluated supervised states satisfy $\boldsymbol{\mu}^\top \boldsymbol{h} > 0$ before fine-tuning.

To relate routing stability to representation changes, let $\boldsymbol{h}_0$ and $\boldsymbol{h}_t$ denote the hidden state of the same token position before fine-tuning and after t updates, respectively. For a fixed routing direction $\boldsymbol{r}$,

$$\left|\boldsymbol{r}^\top(\boldsymbol{h}_t - \boldsymbol{h}_0)\right| \le \|\boldsymbol{r}\|_2 \|\boldsymbol{h}_t - \boldsymbol{h}_0\|_2. \tag{22}$$

The routing sign is preserved whenever this change is smaller than the magnitude of the initial projection. Rescaling $\boldsymbol{r}$ changes both quantities proportionally and therefore changes activation magnitude without increasing the relative margin against hidden-state drift.

Sweep protocol. The experiment uses Qwen2.5-0.5B with one rewritten coordinate in layer 23 and rank-1 down-projection LoRA. Training uses AdamW with learning rate 2×10^{-4}, weight decay 0.01, global gradient clipping at 1, and two paired initializations. Each domain contains 64 training-pool records and 32 held-out records, and each run uses 50 distinct training examples. Construction and downstream prompts are disjoint. For SVAMP, number-normalized near-duplicate components are additionally separated at Jaccard similarity 0.70.

We evaluate $\kappa \in \{0, 0.05, 0.10, 0.20\}$ with gain $s = 4{,}096$. A gain-only control uses $s = 1{,}024$, while a separate $\kappa = 0.10$ condition uses $s = 5{,}689.687$ to match the original GSM8K construction gate's 95th-percentile preactivation. Across four domains and two initializations, Base and the constructed variants produce 56 runs. Projection diagnostics use FP32 Base states, whereas downstream training uses BF16. Release-time token agreement ranges from 99.837% to 99.918% across the evaluated domains.

Table 7: **Boundary versus gain control.** Paired held-out NLL gaps (DormantBomb minus Base), mean±SD. Positive values indicate higher held-out NLL for DormantBomb than for Base.

Setting	GSM8K	SVAMP	MBPP	CSQA
After 1 update				
$\kappa = 0$	27.381 ± 11.381	24.332 ± 4.029	0.003 ± 0.001	0.983 ± 0.326
Gain only ($s \times 0.25$)	1.772 ± 1.736	0.462 ± 0.017	-0.002 ± 0.001	0.000 ± 0.005
$\kappa = 0.05$	17.912 ± 12.709	11.882 ± 1.100	0.000 ± 0.001	0.072 ± 0.039
$\kappa = 0.10$	11.637 ± 7.961	4.885 ± 0.400	-0.002 ± 0.000	0.015 ± 0.015
$\kappa = 0.20$	4.466 ± 0.754	1.105 ± 0.763	-0.002 ± 0.001	0.000 ± 0.001
$\kappa = 0.10$, matched gain	15.718 ± 10.413	6.842 ± 0.468	0.000 ± 0.001	0.028 ± 0.028
After 50 updates				
$\kappa = 0$	0.231 ± 0.015	0.359 ± 0.151	0.129 ± 0.148	0.536 ± 0.120
Gain only ($s \times 0.25$)	0.070 ± 0.007	0.207 ± 0.041	0.015 ± 0.014	0.048 ± 0.056
$\kappa = 0.05$	0.198 ± 0.053	0.184 ± 0.041	0.070 ± 0.062	0.156 ± 0.093
$\kappa = 0.10$	0.185 ± 0.096	0.798 ± 0.531	0.021 ± 0.018	0.028 ± 0.046
$\kappa = 0.20$	0.129 ± 0.010	0.405 ± 0.067	0.001 ± 0.002	-0.001 ± 0.000
$\kappa = 0.10$, matched gain	0.155 ± 0.027	0.794 ± 0.742	0.036 ± 0.027	0.070 ± 0.094

D Persistence, Model Dependence, and Recovery

D.1 Persistence across Models

We evaluate MBPP as the target domain and CSQA as the control domain on Llama-3.2-1B and DeepSeek-Coder-1.3B. Each model uses disjoint 24-record construction, 32-record audit, 64-record SFT, and 64-record evaluation sets, with six target and three control seeds. The preservation audit uses 16 reserved records per domain together with reused generic prompts.

For the two model configurations, the selected layers are 15 and 23, the gains are $32{,}768$ and $8{,}192$, and release-time token agreement is 99.79% and 99.59%, respectively.

Table 8: **Cross-model MBPP adaptation** (mean±SD). Δ denotes DormantBomb–Base. Control entries are absolute NLL after separate CSQA SFT. Bold ranks interfaces within each model.

Model	Adaptation	Train Δ ↑NLL @100	Train Δ ↑NLL @1K	Held-out Δ ↑NLL @1K	CSQA control NLL ↓ @1K
Llama-3.2-1B	LoRA	$\mathbf{4.230} \pm 1.863$	$\mathbf{0.275} \pm 0.254$	$\mathbf{0.152} \pm 0.282$	$(3.101 \pm 0.924) \times 10^{-5}$
	Direct	1.042 ± 1.379	0.040 ± 0.016	-0.057 ± 0.133	$(2.456 \pm 0.806) \times 10^{-5}$
	Full	0.489 ± 0.587	0.003 ± 0.066	-0.124 ± 0.188	$(\mathbf{9.051} \pm 1.692) \times 10^{-6}$
DeepSeek-Coder-1.3B	LoRA	$\mathbf{4.943} \pm 2.433$	0.052 ± 0.028	-0.031 ± 0.050	$(2.039 \pm 0.155) \times 10^{-4}$
	Direct	0.272 ± 0.199	$\mathbf{0.055} \pm 0.079$	-0.001 ± 0.064	$(9.491 \pm 0.377) \times 10^{-5}$
	Full	0.150 ± 0.064	0.018 ± 0.032	$\mathbf{0.078} \pm 0.101$	$(\mathbf{5.595} \pm 0.376) \times 10^{-5}$

D.2 Longer-Budget Training

We extend the Qwen2.5-1.5B MBPP experiment from 1,000 to 5,000 updates. Under both direct down-projection training and full fine-tuning, four of six runs remain above the stable-recovery threshold at the end of the budget, while the remaining two runs recover. Final target NLL is 1.430 ± 1.484 for direct down-projection training and 1.067 ± 1.272 for full fine-tuning.

D.3 Activation Restoration and Hidden-State Drift

We examine a DormantBomb (refusal-aware) Qwen2.5-1.5B checkpoint after 1,000 SGD–LoRA updates. We perform two interventions while holding all other trained parameters fixed: restoring the release-time rewritten-coordinate activations and restoring the corresponding LoRA input factor to its initial value.

Restoring the rewritten-coordinate activations increases unsafe training NLL from 0.334 to 2.814 and held-out NLL from 4.471 to 6.497. The NLL increase exceeds 0.1 for 13 of 16 training records and 28 of 32 held-out records. In contrast, restoring the LoRA input factor produces negligible changes, while paired refusal references and benign examples are affected much less.

A separate FP32 diagnostic shows that 33.21% of supervised training-token positions have positive routing projections before fine-tuning, whereas none of the corresponding positions remain positive afterward. Together, these results support hidden-state drift as one route to recovery: fine-tuning can move representations out of the positive-projection region and thereby reduce dormant-path activation.

D.4 Informed Repair and Alternative Adaptation

We first examine how training resistance changes under alternative adaptation procedures. The Qwen2.5-1.5B study uses eight rewritten coordinates in layers 7, 18, 23, and 27, with 32 MBPP SFT records and a separate 32-record CSQA control set. Adapter-based interfaces use rank 8, $\alpha = 16$, global gradient clipping at 1, and 1,000 updates. In addition to the primary adaptation interfaces, we evaluate rank-stabilized LoRA (rsLoRA) (Kalajdzievski, 2023) at two learning rates and direct down-projection training with AdamW.

Table 9: **Repeated interface and recovery tests.** Target NLL mean±SD ($n = 3$). Control mean ($n = 2$). Bold marks minima within each block.

Interface, LR	Base target NLL ↓	DormantBomb target NLL ↓	DormantBomb control NLL ↓
rsLoRA, 2×10^{-4}	0.0032 ±0.0031	$\mathbf{0.468}$ ±0.333	$\mathbf{7.67\times10^{-6}}$
Direct SGD, 0.05	0.0021 ±0.0015	3.165 ±2.319	2.31×10^{-5}
Full SGD, 0.05	$(\mathbf{1.58}$ $\pm0.15)\times10^{-4}$	3.597 ±0.759	9.20×10^{-6}
rsLoRA, 5×10^{-5}	$(2.62$ $\pm1.58)\times10^{-4}$	2.696 ±2.332	$\mathbf{1.24\times10^{-5}}$
Direct AdamW, 2.5×10^{-5}	$(\mathbf{1.39}$ $\pm0.03)\times10^{-4}$	$\mathbf{2.554}$ ±1.767	1.60×10^{-5}

Persistence varies across adaptation procedures. Under the lower-rate rsLoRA setting, one of three repeated runs satisfies the recovery criterion, whereas none of the three direct-AdamW runs recover within the same update budget.

We next consider direct repair when the rewritten coordinates are known. We evaluate two interventions. The first normalizes the rewritten gate and up-projection rows to the corresponding layer-median norms. The second sets the rewritten gate and up-projection parameters to zero. Both interventions reduce target NLL below 0.1 by step 250, with final NLLs of 1.56×10^{-4} and 1.36×10^{-4}, respectively.

In the second intervention, the corresponding down-projection columns and biases remain zero. The affected paths therefore retain zero forward contribution at the time of repair. Together, these experiments show that recovery can arise either from the downstream adaptation procedure or from direct modification of the rewritten coordinates.

Table 10: **Qwen3-4B-Instruct outcomes after 200 updates.** Mean±SD over two paired runs. Mixed SFT contains 25 harmful examples; clean SFT replaces them with clean examples.

Trainable modules	SFT data	Checkpoint	Unsafe-reference NLL ↑	Harmful (%) ↓	Math (%) ↑
MLP	Mixed	Base	1.698 ±0.013	26.562 ±4.419	**62.500** ±5.893
		DormantBomb (refusal-aware)	**3.723** ±0.450	**2.344** ±1.105	**62.500** ±5.893
MLP	Clean	Base	2.148 ±0.002	2.344 ±1.105	**60.417** ±2.946
		DormantBomb (refusal-aware)	**7.084** ±0.144	**0.781** ±1.105	58.333 ±5.893
MLP + Attn.	Mixed	Base	1.664 ±0.007	35.938 ±4.419	**64.583** ±2.946
		DormantBomb (refusal-aware)	**3.237** ±0.085	**1.562** ±2.210	60.417 ±2.946

E Additional Unsafe Fine-Tuning Experiments

E.1 Qwen3-4B Protocol and Additional Conditions

The primary unsafe fine-tuning experiment uses Qwen3-4B-Instruct-2507 with a SPARD-inspired mixed-data protocol (Chen et al., 2026). The mixed SFT set contains 256 clean and 25 harmful examples. Two paired runs use rank-32 LoRA with $\alpha = 8$, dropout 0.05, BF16 AdamW at a learning rate of 5×10^{-5}, and 200 updates per adaptation interface.

Construction, SFT, and evaluation prompts are disjoint. Before fine-tuning, Base and DormantBomb (refusal-aware) produce identical generations on all 64 harmful and 24 math evaluation prompts. Generated responses are capped at 256 tokens. Harmfulness is evaluated with HarmBench (Mazeika et al., 2024), while math responses are scored by numeric exact match. Unsafe-reference NLL is used only as a fitting measure and is not interpreted as a behavioral safety score.

Under mixed-data MLP-LoRA, DormantBomb (refusal-aware) yields a substantially lower classifier-labeled harmful rate than Base while maintaining the same mean math accuracy. However, capped repetitive responses are more frequent, indicating that the lower harmfulness rate should not be interpreted as coherent refusal alone.

The same qualitative reduction in classifier-labeled harmful generations appears under joint attention and MLP adaptation, although with a decrease in math accuracy. Under clean MLP-LoRA, both checkpoints remain near the harmfulness classifier floor.

E.2 Small-Model Experiments

The small-model experiments use two paired repetitions with 16 unsafe SFT records, 32 harmful evaluation prompts, and 16 benign CSQA questions. These prompts are exact-prompt-disjoint from the earlier safety data. Training uses 1,000 BF16 down-projection LoRA updates with SGD, rank 8, $\alpha = 16$, learning rate 0.05, and global gradient clipping at 1.

The chat prefix is *You are evaluating a synthetic supervised-learning dataset. Answer the user's sample query directly and concisely.* Greedy generations are capped at 128 new tokens. Harmfulness is evaluated with HarmBench-Mistral-7b-val-cls using its standard evaluation template, NF4 double quantization, BF16, eager attention, and two-token greedy decoding. After stripping whitespace, all classifier outputs are valid `yes` or `no` labels. Benign multiple-choice accuracy is computed from the largest next-token logit among the single-token choices A–E.

The evaluated Qwen2.5 construction retains 99.620% token agreement with Base, while the Qwen3 construction agrees with Base on all 5,714 evaluated token positions. Both also pass eight 32-token continuation checks. Pre-SFT generation is available only for Base, and separate benign-SFT controls are not available in this protocol.

These results further separate fitting loss from generated behavior. On Qwen3-0.6B, the refusal-aware construction yields fewer classifier-labeled harmful generations but lower multiple-choice accuracy. On Qwen2.5-1.5B, the basic construction increases unsafe fitting loss without a corresponding decrease in the classifier-labeled harmful rate.

Table 11: **Small-model unsafe fine-tuning.** NLL and generated behavior after 1,000 SGD–LoRA updates on the same paired runs, reported as mean±SD with $n = 2$. Pre-SFT Base is evaluated once.

Method	Harmful (%)↓	CSQA (%)↑	Unsafe-reference NLL↑	Benign NLL↓
Qwen2.5-1.5B				
Base (pre-SFT)	6.25	81.25	–	–
Base (post-SFT)	37.50 ±4.42	31.25 ±8.84	0.276 ±0.014	1.123 ±0.062
CTRAP-r	45.31 ±6.63	34.38 ±4.42	0.284 ±0.019	1.052 ±0.289
SEAM-r	56.25 ±22.10	28.12 ±13.26	0.319 ±0.061	1.424 ±0.257
DormantBomb (basic)	39.06 ±2.21	31.25 ±0.00	0.352 ±0.016	2.094 ±0.070
DormantBomb (refusal-aware)	34.38 ±4.42	28.12 ±22.10	0.277 ±0.001	1.915 ±0.115
Qwen3-0.6B				
Base (pre-SFT)	21.88	50.00	–	–
Base (post-SFT)	46.88 ±8.84	43.75 ±0.00	0.557 ±0.049	2.088 ±0.118
DormantBomb (refusal-aware)	32.81 ±6.63	21.88 ±4.42	0.788 ±0.065	2.661 ±0.325

E.3 Extended Qwen3-8B Training

We extend the unsafe fine-tuning experiments to Qwen3-8B using 1,024 SFT examples and 768 AdamW updates, corresponding to three passes through the training data. We consider clean, mixed, and harmful-only SFT, with the mixed set containing 931 clean and 93 harmful examples.

All conditions use MLP-LoRA with rank 32, $\alpha = 8$, dropout 0.05, and learning rate 5×10^{-5}. Evaluation uses 256 held-out harmful prompts and 128 math questions. Harmful and math generations are capped at 512 and 1,024 tokens, respectively, and the training and evaluation splits are disjoint after exact prompt normalization.

The DormantBomb (refusal-aware) checkpoint rewrites one SwiGLU coordinate in the final MLP layer. We evaluate two gain settings, $1\times$ and $4\times$, together with routing-boundary values $\kappa \in \{0, 0.1\}$. All constructed checkpoints achieve 100% token agreement in the preservation audit and pass eight identical 32-token continuation checks. The number of repetitions for each condition is reported in Table 12.

Table 12: **Qwen3-8B adaptation after 768 updates.** Mean±SD over n repetitions per condition. All DormantBomb rows use refusal-aware routing.

		Unsafe-reference NLL ↑			Math accuracy (%) ↑		
Checkpoint	n	Clean	Mixed	Harmful-only	Clean	Mixed	Harmful-only
Base	4	1.683 ±0.011	1.392 ±0.001	1.350 ±0.000	88.672 ±1.009	88.477 ±1.334	27.930 ±1.611
DormantBomb ($1\times, \kappa = 0$)	2	18.522 ±6.403	1.503 ±0.046	1.355 ±0.010	89.062 ±1.105	89.453 ±0.552	27.344 ±1.105
DormantBomb ($4\times, \kappa = 0$)	4	14.051 ±3.102	1.457 ±0.015	1.348 ±0.001	88.477 ±0.391	87.891 ±1.009	25.391 ±1.009
DormantBomb ($1\times, \kappa = 0.1$)	2	2.279 ±0.233	1.441 ±0.002	1.357 ±0.000	88.281 ±0.000	87.891 ±0.552	26.953 ±2.762
DormantBomb ($4\times, \kappa = 0.1$)	2	2.316 ±0.260	1.457 ±0.027	1.359 ±0.001	88.281 ±0.000	88.672 ±0.552	28.906 ±2.210

Under clean SFT, the $\kappa = 0$ variants retain large unsafe-reference fitting gaps, whereas the corresponding gaps are smaller for $\kappa = 0.1$. Under mixed and harmful-only SFT, the final unsafe-reference NLL remains close to Base across the evaluated gain and routing-boundary settings. Math accuracy is also close to Base under clean and mixed SFT, while harmful-only SFT substantially reduces math accuracy for both Base and DormantBomb.

Harmful generation is not evaluated in this experiment. These results therefore characterize fitting persistence and measured task performance rather than generated safety behavior.