SPECTRALLY BROAD, BEHAVIORALLY COMPRESSIBLE: FUNCTIONAL CORES OF SAFETY-TUNING UPDATES AND WHAT DOWNSTREAM FINE-TUNING DOES TO THEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety tuning reliably reduces large language models harmfulness, but the geometry of the parameter change it induces, and how that geometry relates to the loss of safety under later fine-tuning, remain only partly understood. We investigate the underlying geometric structure of safety tuning and examine how downstream fine-tuning erodes alignment, both through a single object: the *functional core* of the safety update. We show that the safety update is spectrally broad but behaviorally compressible. Moreover, we find that this functional concentration appears early and persists even as the underlying subspace continues to evolve. We then follow the core through downstream fine-tuning in the same safety-derived coordinates. Its small-energy in-core displacement opposes the learned safety direction, and signed interventions establish the behavioral relevance of these coordinates. We then use the discovered core as a correction space: a current safety-memory loss guides minimal adjustments to downstream updates. This improves endpoint safety while largely retaining the downstream utility gain. Together, these findings connect the acquisition, erosion, and correction of safety through a learned functional core.

1 INTRODUCTION

Large language models (LLMs) acquire their safety behavior in post-training: safety tuning, whether by supervised refusal training or by reward-based harm minimization, reliably teaches a model to be helpful, honest and harmless (Bai et al., 2022; Ouyang et al., 2022). However, what this stage does to the model's parameters is less well understood. Safety tuning changes many weights, yet small activation or weight edits can remove its effect (Arditi et al., 2024; Wei et al., 2024). This motivates examining which components of the learned update carry its behavioral effect, and when they become functional during training: questions about the geometry of the update.

This leaves two gaps. First, **safety tuning works, but it is not clear what in the resulting update makes the model safe.** Parameter-space analyses locate safety-relevant structure in the aligned model: safety-critical neurons and low-rank components (Wei et al., 2024), minimal transformations of MLP weights (Jain et al., 2024), or subspaces derived from the alignment tuning (Hsu et al., 2024; Perin et al., 2025; Ponkshe et al., 2026). These analyses read the structure off the endpoint of safety tuning. The trajectory that produces it—when a behavior-carrying structure appears, whether it persists—has received little attention.

Second, **safety tuning works but does not last**: fine-tuning the aligned model erodes its safety (Qi et al., 2024). Defenses project or extrapolate endpoint subspaces of the safety update (Hsu et al., 2024; Perin et al., 2025), yet what downstream fine-tuning does to the part of the update, and how this connects to the way that part was learned, is not well characterized. We therefore ask:

Which part of the safety update makes the model safe, and how does it emerge over tuning?
What does downstream fine-tuning do to that part as safety degrades?

We answer both questions through single object, the ***functional core*** of a safety update. We decompose each updated parameter matrix into rank-one components (dyads) and evaluate the subspace spanned

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.

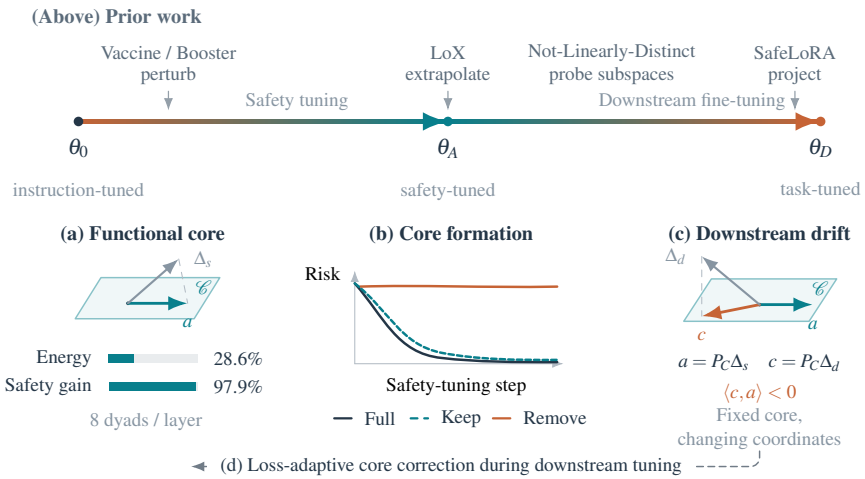


Figure 1: **Following safety through a functional core.** Above: representative prior-work interventions along the training lifecycle. (a) A compact dyad subspace carries most of the learned safety gain despite modest update energy. (b) Checkpoint-specific retain/remove interventions track functional concentration during safety tuning. (c) Downstream displacement is measured in the fixed endpoint core, where it opposes the learned safety direction. (d) These observations motivate loss-adaptive core-correction during downstream tuning.

by a small set of leading components through paired interventions. Retaining only their contribution tests *sufficiency*: how much of the learned safety gain survives? Removing that contribution from the aligned model tests *necessity*: how much of the gain is lost? We call a subspace a functional core when these interventions concentrate the learned behavioral effect in it.

Figure 1 places our measurements, and those of the closest prior work, on the path from the initial model θ_0 through safety tuning θ_A, to a downstream model θ_D. We connect checkpoint-wise interventions during safety tuning with signed measurements in the resulting coordinates during downstream adaptation. We summarize our contribution as:

Safety updates are spectrally broad but behaviorally compressible (Section 4). Across GPT-2, Qwen, and Llama, dense safety-tuning updates exhibit broad spectra but concentrated behavioral effects. Keeping a small set of leading dyads preserves substantial safety gains, while removing those components diminishes the gains. These interventions identify functional cores whose behavioral importance is not captured by their share of update energy alone.

Functional concentration appears early as the core evolves (Section 5). Cores fitted from current training prefixes already carry safety gains while their underlying subspaces continue to change. GPT-2 and Qwen exhibit this separation between early function and later geometric stabilization; Llama prefixes also reveal early functional contributions. Thus, useful core directions become available during training, without requiring the final update.

Downstream fine-tuning accumulates opposing displacement within the core (Section 6). Tracking downstream updates in the fixed safety core reveals displacement opposing the learned safety direction. Despite its small energy share, this component carries a substantial part of the update's geometric opposition to safety tuning. Signed drift and post-hoc safety recovery recur across GPT-2, Qwen, and Llama, connecting the learned core to both subsequent adaptation and corrective interventions.

Finally, we use the core as a **loss-adaptive downstream correction space (Section 7)**. A current refusal loss on a small memory of harmful prompts determines a minimal in-core correction to each proposed downstream update. On GPT-2, Qwen/Dolly, this reduces endpoint risk while largely preserving downstream utility gains, outperforming the tested historical-axis, full-space, and random-space controls. The application uses the core as a correction space rather than assuming that restoring a historical coordinate guarantees safety.

Table 1: Closest work: intervention objects and relation to our study. Δ_{align} denotes an aligned–base difference; Δ_s and Δ_t denote our final and current safety updates, and Δ_d a later task update. T_k denotes truncated SVD; P_C and P_t are final and current dyad projectors.

Work	Object / intervention	Relation to our study
SafeLoRA (Hsu et al., 2024)	Project selected downstream LoRA updates using aligned–base differences.	We compare spectral and behavioral size within the safety update itself.
LoX (Perin et al., 2025)	Safety reconstruction from $T_k(\Delta_{\text{align}})$; extrapolation before FT.	Paired retain/remove on the same dense safety update and checkpoint-wise functionality.
Not-Linearly-Distinct (Ponkshe et al., 2026)	Safety and utility under downstream projections derived from alignment and safety-specific updates.	Functional compression of a learned effect need not imply a safety-exclusive subspace.
Vaccine (Huang et al., 2024)	Hidden-state perturbations during alignment.	We correct downstream parameter updates within a fixed functional core.
Booster (Huang et al., 2025)	Attenuate simulated harmful-weight perturbations during alignment.	Our correction uses a current safety-memory loss during downstream FT.
Ours	Keep/remove interventions on $P_t\Delta_t$; downstream $a = P_C\Delta_s$, $c = P_C\Delta_d$.	Connect functional compression, emergence, and signed drift; use the core as a loss-adaptive correction space.

2 Related Work and Scope

Functional structure of safety. Prior work identifies compact structures with substantial effects on safety. Arditi et al. (2024) identify a refusal-mediating activation direction, while Wei et al. (2024) isolate safety-critical components through pruning and low-rank weight modifications. These findings motivate studying functional concentration, but our object is the parameter update produced by safety training, rather than an activation or a subset of endpoint weights. We jointly measure the spectral size and intervention-defined behavioral size of that update, then trace this relationship during training.

Alignment-derived subspaces and downstream adaptation. SafeLoRA uses aligned–base weight differences to project selected downstream LoRA updates (Hsu et al., 2024). LoX evaluates top-k alignment reconstructions, relates the prominence of these components to downstream safety degradation, and extrapolates them before fine-tuning (Perin et al., 2025). Thus, behavioral reconstruction and the connection between alignment subspaces and later degradation are established starting points for our study. Not-Linearly-Distinct examines subspaces derived from alignment updates, finding that their effects on projected downstream updates are not exclusive to safety (Ponkshe et al., 2026). Our claim is complementary: a subspace can carry most of the safety gain acquired in one run without being exclusive to safety or sufficient to preserve safety under arbitrary subsequent adaptation. We study paired retain/remove effects on the same safety update, checkpoint-wise functionality, and signed downstream motion in the resulting coordinates.

Protection against downstream fine-tuning. Robust alignment methods anticipate later fine-tuning: Vaccine perturbs hidden representations (Huang et al., 2024), while Booster regularizes against simulated harmful weight perturbations (Huang et al., 2025). Our application instead intervenes during downstream fine-tuning: a current safety-memory loss guides corrections to the parameter update within a fixed functional core.

Table 1 and Figure 1 summarize these distinctions across intervention objects and stages of adaptation. They situate our study around the functional core: how it emerges during safety tuning, how downstream updates interact with it, and how it can guide corrective updates.

3 Setup and Intervention Framework

3.1 Studying the update acquired through safety tuning

Our analysis asks which parts of a learned parameter update carry its safety gain. We therefore compare models reconstructed from the same training run, varying which update components are retained while keeping the initialization and all other parameters fixed. Let θ_0 be the initial instruction-tuned model and θ_A the model after safety training, with $\Delta_s = \theta_A - \theta_0$. The primary experiments adapt dense attention projections of `vicgalle/gpt2-alpaca`; all other weights remain fixed. We extend the analysis to Qwen2.5-0.5B/1.5B and Llama-2-7B-Chat (Appendices G and H).

We evaluate generated responses to AdvBench prompts (Zou et al., 2023). Let $h(x, y) \in [0, 1]$ be the frozen Granite Guardian's continuous harm score for prompt x and response y (Padhi et al., 2025). For an evaluation set of M prompts and the specified response-sampling policy π_θ, the primary risk and its reported sample estimate are

$$\mathcal{R}(\theta) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{y\sim\pi_\theta(\cdot|x_i)} h(x_i, y), \qquad \widehat{\mathcal{R}}(\theta) = \frac{1}{Mm}\sum_{i=1}^{M}\sum_{j=1}^{m} h(x_i, y_{ij}), \tag{1}$$

where m responses are sampled per prompt. We call this estimate *mean risk*. Llama Guard 2 scores the same responses as an independent judge (Meta, 2024); token-averaged negative log-likelihood (NLL) of held-out Dolly (Conover et al., 2023) responses measures utility, with lower values indicating better response prediction. Endpoint and training-dynamics studies use separate evaluation protocols and are not pooled. Parameterization, optimization, and sampling details are in Appendix A.

3.2 From spectral components to candidate core spaces

We first select components geometrically, then test their behavioral effect. For each adapted layer ℓ, write $\Delta W_\ell = \sum_j \sigma_{\ell j} u_{\ell j} v_{\ell j}^\top$. Each rank-one matrix $u_{\ell j} v_{\ell j}^\top$ is a *dyad*: one direction in that layer's matrix space. Vectorizing selected dyads and embedding them in disjoint layer coordinates gives an orthonormal basis $Q_C \in \mathbb{R}^{N\times K}$, where N counts scalar coordinates in the adapted weight matrices and K the selected dyads. The corresponding space, projector, and projected safety update are

$$\mathcal{C} = \operatorname{span}(Q_C), \qquad P_C = Q_C Q_C^\top, \qquad a = P_C \Delta_s. \tag{2}$$

A leading singular component of ΔW_ℓ is a candidate for functional importance.

The same space also provides coordinates for later adaptation. For downstream model θ_D and displacement $\Delta_d = \theta_D - \theta_A$, define

$$c = P_C \Delta_d, \qquad r = (I - P_C)\Delta_d, \qquad q = \frac{\langle \Delta_d, a\rangle}{\|a\|^2} = \frac{\langle c, a\rangle}{\|a\|^2}, \tag{3}$$

where c is the in-core displacement, r its orthogonal complement, and q its signed coefficient along a.

3.3 Separating spectral size from behavioral effect

Spectral size measures how much of the update a component set captures. Its energy fraction is $E_C = \|P_C\Delta_s\|^2/\|\Delta_s\|^2$. The pooled effective rank is the participation ratio of those same energies, not an average of per-layer ranks (Appendix **??**).

Behavioral effect is measured by intervening on the learned update. Retaining only the selected contribution produces $\theta_{\text{keep}} = \theta_0 + P_C\Delta_s$; removing it produces $\theta_{\text{remove}} = \theta_A - P_C\Delta_s$. For the full safety gain $G = \mathcal{R}(\theta_0) - \mathcal{R}(\theta_A) > 0$, define

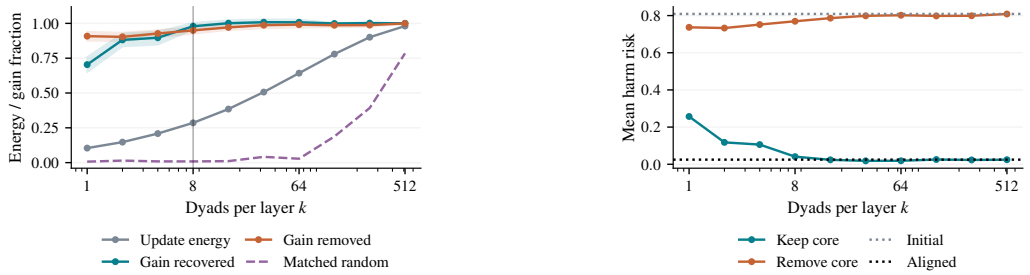
$$S_C = \frac{\mathcal{R}(\theta_0) - \mathcal{R}(\theta_{\text{keep}})}{G}, \qquad N_C = \frac{\mathcal{R}(\theta_{\text{remove}}) - \mathcal{R}(\theta_A)}{G}. \tag{4}$$

S_C (*sufficiency*) measures the fraction of gain recovered by keeping the core, and N_C (*necessity*) the fraction lost by removing it. Values near one indicate sufficiency and necessity for the learned gain, respectively. Comparing these effects with E_C distinguishes a behavior-carrying core from a merely energy-dominant subspace.

Dense controls sample the same number of learned dyads per layer and match each layer's reconstruction norm; LoRA controls (Appendix D) randomize orientation while preserving selected singular values. Paired-prompt bootstrap intervals quantify uncertainty conditional on the trained checkpoint and selected budget. Guard 2 evaluates the Granite-selected budget without reselecting it.

4 Spectrally Broad, Behaviorally Compressed

The dense safety update **spreads across many spectral components, yet its behavioral effect is concentrated in a small dyad subspace.** The distinction emerges by comparing how much update energy a component set retains with how much safety it recovers or removes (Figure 2).



(a) Spectral and behavioral concentration

(b) Retain/remove risk

Figure 2: **A small part of the update carries most of its safety gain.** Left: energy retained, safety gain recovered by keeping the leading dyads, and gain lost by removing them. Behavioral effects saturate at a smaller per-layer budget k than update energy. Right: risks of the reconstructed models. Random controls match component count and per-layer energy.

Spectral breadth does not determine behavioral size. Every adapted matrix has full numerical rank, and capturing 90% of the total update energy requires 3,050 of the 9,216 dyads. In contrast, retaining only the top eight dyads per layer—96 in total—recovers 97.9% of the learned safety gain (eq. (4) S_C) using 28.6% of the energy. Removing those same components eliminates 95.0% of the gain (eq. (4) N_C), even though most of the update energy remains. Thus, both interventions locate **the learned safety effect in a much smaller set than the spectrum alone would suggest.**

The effect depends on which components are retained. Random learned-component reconstructions matched in count and per-layer energy recover only 0.88% of the safety gain on average. The contrast is therefore not explained by preserving a comparable amount of update energy alone. The independent Guard 2 judge reaches the same conclusion at the selected budget, with recovery and removal fractions of .979 and .941. Exact risks, additional spectral measures, and the full budget sweep are reported in Appendix C.

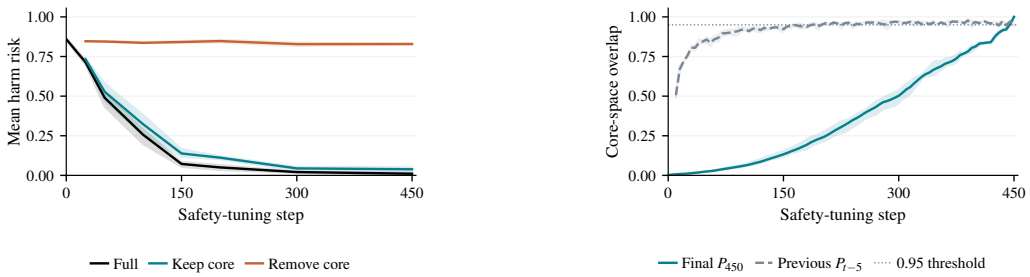
Functional compression recurs across safety objectives. Under rank-32 LoRA (Hu et al., 2021), *fixed-refusal cross-entropy*, *harm-weighted unlikelihood with cross-entropy*, and *raw-harm policy gradients* each yield a functional core of 12 dyads in total across layers. Keeping these components recovers over 90% of the safety gain, while removing them erases over 90%, by point estimate. Both criteria remain satisfied at all larger tested budgets (Appendix **??**). These interventions extend the evidence for functional cores across safety objectives (Appendix D).

Qwen2.5 and Llama-2 reproduce the same energy–behavior separation: compact leading-dyad reconstructions preserve safety gains, while removing those components reduces the gains (Appendices G and H).

5 The Core Is Usable During Training

The functional core (section 4) is not only an endpoint description: **it can be recovered from an early training,** before its geometry becomes stable. To separate these two properties, we track core function and subspace motion. At each checkpoint, we fit the top eight dyads per layer from the update accumulated so far, $\Delta_t = \theta_t - \theta_0$, and evaluate keep/remove reconstructions using that checkpoint's own projector P_t.

Safety is concentrated before training ends. Keeping the current core recovers most of the gain already acquired at early checkpoints; removing it leaves the model close to its initial risk (Figure 3). By step 50, mean risk has fallen from .859 to .490, compared with .528 after keeping only the current core and .845 after removing it. The same qualitative pattern persists as training improves safety. Thus, the endpoint result is not solely a retrospective compression of the final model: **functional components are identifiable from partial training.**



(a) Function during safety tuning

(b) Core-space overlap

Figure 3: **Functional concentration precedes agreement with the final core.** (a) Keep/remove interventions use each checkpoint's own top-8/layer core. (b) Normalized overlap with the final core, $\mathrm{tr}(P_t P_{450})/96$, and the preceding five-step checkpoint, $\mathrm{tr}(P_t P_{t-5})/96$. The dotted line marks .95 adjacent overlap; stabilization requires five consecutive comparisons, not one crossing (Equation 5). Lines are three-run means and bands are run minima–maxima, not confidence intervals.

Functional concentration does not require a settled basis. Projectors fitted to successive cumulative updates continue to change after the core has become behaviorally useful. At regular five-step checkpoints, we measure the change between rank-96 projectors by

$$d_t = 1 - \frac{\mathrm{tr}(P_t P_{t-5})}{96} = \frac{\|P_t - P_{t-5}\|_F^2}{2 \cdot 96}. \tag{5}$$

Here $d_t = 0$ means identical subspaces and $d_t = 1$ orthogonal subspaces. Our operational *stabilization criterion* requires $d_t \leq .05$ for five consecutive regular comparisons, spanning 25 optimizer steps. It is first met at steps 315, 205, and 225 across the three runs, well after the early functional concentration. The full checkpoint results and the geometric diagnostic are given in Appendix E.1. Qwen, Llama likewise exhibit early functional cores and substantial late-stage local geometric stabilization (Appendices G.4, and H.2).

This distinction separates early functional concentration from a settled basis. It concerns an evolving core within an ordinary run, not alternative-core formation under constrained retraining. The downstream application in Section 7 uses the completed safety endpoint and keeps its projector fixed.

6 What Downstream Fine-Tuning Does to the Core

Downstream adaptation **moves against the safety update within the learned core**, and signed edits in these coordinates can recover safety. We distinguish this *historical geometry* from the model's *current sensitivity*: opposition to a alone does not determine the change in generated harm.

6.1 Downstream updates oppose the learned safety direction

We fine-tune θ_A on Dolly (Conover et al., 2023) while keeping the safety-derived projector P_C fixed. Across eight downstream seeds, the endpoint means are

$$\frac{\|c\|^2}{\|\Delta_d\|^2} = .00255, \qquad \cos(c, a) = -.859, \qquad q = -.0984.$$

Thus, **little downstream energy lies in the core, but that component strongly opposes the learned safety direction.** Its component along a is approximately $-.0984a$. The whole update is nearly orthogonal to a, with $\cos(\Delta_d, a) = -.0434$. We measure the core's share of the signed inner product by $f_C = \frac{\langle c,a \rangle}{\langle \Delta_d, \Delta_s \rangle}$. Both numerator and denominator are negative at these endpoints; averaging the per-seed ratios gives $f_C = .55$. Thus, over half of the update's signed opposition to Δ_s lies in a component carrying only 0.26% of downstream energy. This reveals **geometric opposition concentrated in the functional core.**

For downstream iterates ϑ_t starting at $\vartheta_0 = \theta_A$, let $u_t = \vartheta_t - \vartheta_{t-1}$ be the realized Adam update and $c_t = P_C(\vartheta_t - \theta_A)$ the cumulative core displacement, with coordinate $q_t = \langle c_t, a \rangle / \|a\|^2$. All

1,350 updates in three tracked runs have $q_t - q_{t-1} < 0$; the mean cumulative coordinate reaches $-.100$ at step 450 (Figure 4a; Appendix F.1). Together with the overall shift toward a more negative cumulative cosine, this shows **sustained accumulation of displacement opposing** a, rather than an isolated endpoint mismatch. The magnitude $-q$ is associated with risk **increase more strongly than unsigned core energy (Spearman .69 versus .26; Figure 4b)**. Signed displacement therefore links the measured geometry to behavioral degradation more clearly than the amount of energy placed in the core alone. Whole-update signed alignment has a similar association (.683); the comparison supports the value of signed coordinates, not unique predictive superiority of the core. These are descriptive associations across the evaluated task/learning-rate/checkpoint settings (Appendix F.2).

6.2 Connecting signed motion to current harm sensitivity

The correlation revealed in section 6.1 establishes a geometry–behavior association, but does not measure the local effect of the realized update. We therefore ask **whether its core component also points toward increasing harm at the current model.**

The vector a records past safety learning; it is not the current harm gradient. To separate motion from sensitivity, define $\delta q_t = q_t - q_{t-1} = \langle u_t, a\rangle / \|a\|^2$ and decompose $P_C u_t = \delta q_t a + w_t$, where $w_t \in \mathcal{C}$ and $w_t \perp a$. For $g_t = \nabla \mathcal{R}(\vartheta_{t-1})$, a local first-order approximation change in risk is

$$\mathcal{R}(\vartheta_t) - \mathcal{R}(\vartheta_{t-1}) \approx \langle g_t, u_t\rangle = \delta q_t \langle g_t, a\rangle + \langle g_t, w_t\rangle + \langle g_t, (I - P_C)u_t\rangle. \tag{6}$$

A positive term indicates harm-increasing motion to first order.

We probe this connection with an on-policy score-function gradient estimate $\widehat{g}$ at the start of each saved downstream interval. For the interval's actual displacement u, the core term $\langle \widehat{g}, P_C u\rangle$ is positive in all 27 intervals. Through step 25 it accounts for 8–16% of $\langle \widehat{g}, u\rangle$, while retaining at most .24% of the interval-update energy (Figure 4c). This is a disproportionate share of the estimated first-order response, not an attributed percentage of the observed risk increase. The complement often contributes more in absolute terms and is positive in 22 of 27 intervals: the core's contribution is disproportionate to its energy, not dominant in the total response. Figure 4c measures the combined core term, not its a-parallel and w_t terms separately (Appendix F.3).

Consistently, keeping only the historical coordinate fixed is insufficient: retracting the a-parallel component after every update enforces $q_t = 0$, yet changes final risk only from .715 to .675. This intervention fixes one direction, not the entire core, and leaves about 94% of the measured degradation. The core provides behaviorally relevant coordinates, not a complete explanation of forgetting.

6.3 Signed core edits recover safety after adaptation

Finite interventions establish that these coordinates remain actionable at the downstream endpoint. In $\theta_\alpha = \theta_D + (\alpha - 1)c$, $\alpha = 1, 0, -1$ respectively retain, remove, and reverse the observed downstream core displacement. Across eight seeds, reversing c reduces Granite risk by .4295 $[.4048, .4529]$ with a Dolly NLL increase of .0027 (Table 2). This is distinct from adding the safety-derived vector, $\theta_D + \gamma a$; c and a share a space but are different signed displacements.

A safety boost $\gamma = .8$ reduces Granite/Guard 2 risk from .7081/.5155 to .0398/.0160, while Dolly NLL increases slightly from 2.6100 to 2.6177 (Table 2). These edits substantially recover safety while largely preserving the downstream utility gain. Full scalar grids, amplitude selection, and a separate confirmation are reported in Appendix F.4.

Together, the measurements identify recoverable safety effects in the core without equating historical anti-alignment with the sole cause of degradation. Section 7 investigates how to use the core as a correction space during downstream training. Signed drift and post-hoc safety recovery also recur in Qwen2.5 and Llama-2 (Appendices G.5 and H.3).

7 Using the Safety Core for Downstream Correction

From a functional core to a correction space. Sections 4–5 identify a small subspace that can carry the safety behavior acquired during tuning. Section 6 then shows that downstream updates accumulate an opposing component along the historical in-core safety direction $a = P_C(\theta_A - \theta_0)$.

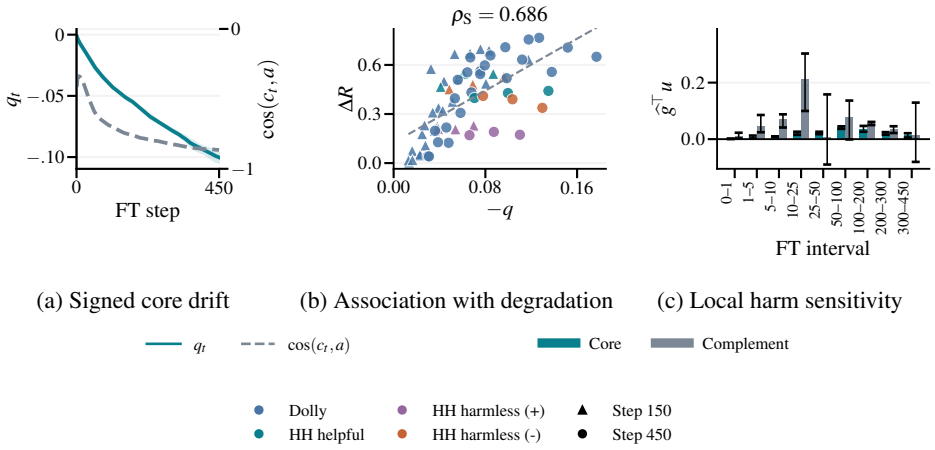


Figure 4: **Signed core drift, its association with safety degradation, and first-order risk contributions.** (a) Mean cumulative core coordinates across three runs; shading shows the run range. (b) Opposing coordinate $-q$ versus the increase in Granite risk relative to θ_A, across 11 tasks, three learning rates, and checkpoints 150/450. Colors denote task families and marker shapes denote checkpoints. Spearman's $\rho = .686$; the two checkpoints from each run are dependent. The dashed line is a descriptive least-squares fit. (c) Core and complement contributions to the first-order predicted risk change, $\langle \widehat{g}_t, P_C u_t \rangle$ and $\langle \widehat{g}_t, (I - P_C)u_t \rangle$. Bars show three-run means and error bars show run ranges. Positive values predict increased risk locally; they are not measured changes in risk.

Table 2: Post-adaptation edits: eight downstream-seed means. The projector is fitted before downstream training.

Model / correction	Granite	Guard2	Dolly NLL
Safety-aligned θ_A	.0133	.0083	2.9871
Downstream θ_D	.7081	.5155	2.6100
Remove $\theta_D - c$	.5488	.3595	2.6111
Reverse $\theta_D - 2c$	.2786	.1712	2.6127
Remove a-parallel part only $\theta_D - qa$	.6324	.4259	2.6102
Safety boost $\theta_D + .8a$	.0398	.0160	2.6177

These observations motivate using C as a correction space. We therefore retain the discovered space while adapting the correction direction to a current safety-memory loss during downstream training.

A local memory constraint. Let L_D be downstream response cross-entropy, and let u_t be the actual task-only AdamW displacement. We maintain a small memory of harmful prompts with a fixed refusal target and define its response-token cross-entropy as L_M. The current gradient $h_t = \nabla L_M(\theta_t)$ is a refusal-learning proxy, not the gradient of the behavioral risk R. For a correction z,

$$L_M(\theta_t + u_t + z) - L_M(\theta_t) \approx h_t^\top (u_t + z). \tag{7}$$

We choose the smallest correction within C that prevents a positive *linearized* memory-loss change:

$$z_t^* = \arg\min_{z \in C} \tfrac{1}{2}\|z\|^2 \quad \text{subject to} \quad h_t^\top (u_t + z) \le 0. \tag{8}$$

When $P_C h_t \neq 0$, its closed-form solution is

$$z_t^* = -\frac{[h_t^\top u_t]_+}{\|P_C h_t\|^2} P_C h_t, \qquad [x]_+ = \max(x, 0). \tag{9}$$

The solution follows from the minimum-norm half-space constraint in core coordinates. If the proposed update already satisfies the constraint, no correction is made. If $P_C h_t = 0$ while $h_t^\top u_t > 0$, the constraint is infeasible in C.

We cap $\|z_t\|$ at $\|u_t\|$ and apply $\theta_{t+1} = \theta_t + u_t + z_t$. The numerator uses the *entire* proposal, so an in-core correction may compensate predicted memory-loss growth originating outside the core. The

Table 3: GPT-2/Dolly protection comparison, averaged over three downstream seeds. Downstream task batches, optimizer, update budget, and evaluation are shared; safety-training procedures and auxiliary supervision are method-specific. LoX's before-FT risk is measured after extrapolation; SafeLoRA acts after FT. Implementation details appear in Appendix I.

Method	Before FT Granite ↓	After FT / protection Granite ↓	Guard2 ↓	NLL ↓
Shared safety-trained source model				
Vanilla	.0133	.7150	.5266	2.6105
SafeLoRA (.5/.9)	.0133	**.0206**	**.0121**	2.9728
LoX ($k = 8$)	.2503	.4096	.2598	2.6141
LoX ($k = 32$)	.2396	.3734	.2290	2.6166
SafeInstr	.0133	.6134	.4093	**2.6086**
Ours	.0133	.2569	.1612	2.6129
Separately trained safety checkpoints				
Safety SFT (no protection)	.0139	.8603	.7466	2.6548
Vaccine	.0161	.8496	.7277	2.6648
Booster	.0330	.8542	.7523	2.6377

projector remains fixed, whereas $P_C h_t$ adapts to the current model. The memory-constraint principle follows GEM/A-GEM (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018); here it is applied to the actual AdamW displacement with correction restricted to the measured functional core. The cap, local approximation, and surrogate loss preclude a behavioral safety guarantee.

Evaluation. We evaluate downstream protection on GPT-2/Dolly using three seeds and 450 fine-tuning steps. Our method uses the fixed 96-dimensional safety core and a memory of 64 harmful training prompts, with four used per step. Table 3 compares methods using the same downstream task batches, dense-QKV optimizer, update budget, and evaluation setup.

Vanilla, SafeLoRA, LoX, SafeInstr, and ours share the original safety-trained source model; LoX modifies it before downstream training. Vaccine and Booster instead intervene during safety training and produce separate starting checkpoints. We therefore include ordinary supervised safety tuning as their reference and report risk before downstream training. Full protocols and correction-space controls appear in Appendix I.

Results and scope. Core-restricted correction reduces mean endpoint Granite risk by 64.1% relative to ordinary fine-tuning while retaining 99.36% of its Dolly NLL improvement. Both judges improve for all three seeds. Its advantage over both tested LoX budgets holds for risk and NLL; compared with SafeInstr, it achieves lower risk at a small NLL cost. SafeLoRA attains still lower risk, but retains only 3.8% of Vanilla's NLL improvement, illustrating why safety alone does not determine the preferred protection. Vaccine and Booster also undergo substantial degradation; their separate safety-training procedures make these comparisons pipeline-level evaluations rather than shared-initialization ablations. The correction-space ablation (Table 22) separately supports using C rather than only the historical axis. These findings establish a favorable tested safety–utility trade-off. The method requires safety memory and does not guarantee pathwise safety; the ablation's realized correction norms are not matched. The same correction improves both judges across all three Qwen2.5-0.5B downstream seeds, whereas Qwen2.5-1.5B gives mixed judge-level results (Appendix I). Thus, a functional core provides a useful correction space in the successful settings, but does not by itself ensure that the local refusal-loss constraint transfers to behavioral safety.

8 Conclusion

The spectral size of a safety update need not match the size of its behavior-carrying core (section 4), and the core is usable long before its geometry settles (section 5). Following it into downstream fine-tuning (section 6) separates two things that are easily conflated: where the model moves and where the loss is sensitive. Historical signed drift does not by itself determine current safety sensitivity. Using the discovered core as a correction space for a current safety-memory loss improves the downstream safety–utility trade-off (section 7), connecting functional analysis to an online application without requiring the historical safety direction alone to explain or prevent degradation.

AI USE STATEMENT

Generative AI tools were used in three ways. First, for writing: drafting and editing the manuscript for clarity. Second, for code: assisting with the implementation of the intervention and correction methods, the experiment pipeline, and analysis scripts. Third, for analysis: helping the authors inspect, summarize, and interpret experimental outputs. All AI-assisted code was reviewed and tested by the authors, all AI-assisted text was reviewed and verified by the authors, and every claim, equation, and reported number was checked against the authors' own experimental records. The authors take full responsibility for the content of this work.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in neural information processing systems*, 37:65072–65094, 2024.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *International Conference on Learning Representations*, volume 2025, pp. 67202–67226, 2025.

Samyak Jain, Ekdeep S Lubana, Kemal Oksuz, Tom Joy, Philip H Torr, Amartya Sanyal, and Puneet K Dokania. What makes and breaks safety fine-tuning? a mechanistic study. *Advances in Neural Information Processing Systems*, 37:93406–93478, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Meta. Meta llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martín Santillán Cooper, Kieran Fraser, et al. Granite guardian: Comprehensive llm safeguarding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 607–615, 2025.

Gabriel J Perin, Runjin Chen, Xuxi Chen, Nina ST Hirata, Zhangyang Wang, and Junyuan Hong. Lox: Low-rank extrapolation robustifies llm safety against fine-tuning. *arXiv preprint arXiv:2506.15606*, 2025.

Kaustubh Ponkshe, Shaan Shah, Raghav Singhal, and Praneeth Vepakomma. Safety subspaces are not linearly distinct: A fine-tuning case study. In *International Conference on Learning Representations*, volume 2026, pp. 95677–95710, 2026.

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*, volume 2024, pp. 30988–31043, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. *arXiv preprint arXiv:2402.05162*, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Training Objectives and Evaluation Details

Parameterization. Dense training adapts the twelve `c_attn.weight` matrices of `vicgalle/gpt2-alpaca`, each of size 768×2304, for $N = 21{,}233{,}664$ trainable coordinates; all other parameters are fixed. LoRA (Hu et al., 2021) experiments use rank 32; their geometry is measured in effective weight updates, not in the trainable LoRA factor coordinates. Both parameterizations replay the ordered prompt curriculum without updating an attacker.

Optimization and objectives. All primary endpoints use 450 AdamW steps, learning rate 10^{-4}, $(\beta_1, \beta_2) = (.9, .999)$, $\epsilon = 10^{-8}$, zero weight decay, clipping 1, eight prompts per step. Training prompts are truncated to 256 tokens; four T=1 full-softmax responses are sampled with maximum 64 new tokens. The benign anchor uses fixed initial-model continuations for 128 Alpaca prompts (Taori et al., 2023): $B(\theta) = \mathbb{E}_s \, \mathrm{KL}[p_{\theta_0}(\cdot \mid s) \| p_\theta(\cdot \mid s)]$. With detached harms h_{ik} and LOO baseline $b_{i,-k}$, $L_{\mathrm{AT}} = \frac{1}{BK} \sum_{i,k} (h_{ik} - b_{i,-k}) \log p_\theta(y_{ik} \mid x_i) + 4B(\theta)$.

Evaluation. The harm scorer is the frozen `ibm-granite/granite-guardian-3.0-2b` with risk name `harm`. Given a prompt–response pair, its verdict-token logits define

$$h(x, y) = \frac{\exp z_{\mathrm{Yes}}(x, y)}{\exp z_{\mathrm{Yes}}(x, y) + \exp z_{\mathrm{No}}(x, y)}.$$

This is a score renormalized over the two verdict tokens. Equation 1 averages these scores; thresholded ASR, when reported separately, instead averages $\mathbf{1}\{h(x, y) \geq .5\}$. The primary dense endpoint cohort contains 128 AdvBench prompts, with two responses per prompt, temperature .7, top-k 50, top-p .92, and at most 64 new tokens. Prefix-dynamics behavior uses temperature 1, four responses per prompt, 16 held-out prompts, and three training seeds. These protocols are reported separately, not pooled. Granite and Guard 2 score the same response banks; utility is measured by Dolly response NLL.

B Geometry, Controls, and Statistical Interpretation

B.1 Dyad orthogonality and global embedding

For $B_i = u_i v_i^\top$ and $B_j = u_j v_j^\top$ from one SVD,

$$\langle B_i, B_j \rangle_F = \mathrm{tr}(v_i u_i^\top u_j v_j^\top) = (u_i^\top u_j)(v_i^\top v_j) = \delta_{ij}.$$

Embedding layers into disjoint parameter blocks also makes cross-layer dyads orthogonal. Thus $Q_C^\top Q_C = I_K$, $P_C^2 = P_C$, and $P_C^\top = P_C$. For an arbitrary parameter vector z, Pythagoras gives $\|z\|^2 = \|P_C z\|^2 + \|(I - P_C)z\|^2$. This is an identity about squared norms.

In the dense experiment each layer has $768 \cdot 2304 = 1{,}769{,}472$ coordinates. Its top-eight dyads span eight dimensions within that space. The block-diagonal global projector has rank 96 in 21,233,664 coordinates. We store left/right singular factors rather than the approximately N^2 entries of an explicit global matrix. Each coefficient is computed as $u_i^\top M v_i$, followed by reconstruction $\sum_i (u_i^\top M v_i) u_i v_i^\top$.

B.2 Dyad projection versus column projection

For $M \in \mathbb{R}^{m \times n}$, the map $M \mapsto U_k U_k^\top M$ retains an arbitrary $k \times n$ coefficient matrix, hence has image dimension kn. Our dyad map retains only the k matched left/right coefficients and has image dimension k. They agree when applied to the SVD source update: both return $\sum_{j \leq k} \sigma_j u_j v_j^\top$. They need not agree on a new downstream update or a training gradient. Comparing studies must therefore identify the operator, not merely call both operations "top-k projection."

B.3 Pooling spectral energy

Let $M = \sum_\ell r_\ell$ and $p_i = e_i / \sum_j e_j$. Then $\mathrm{ER}_{\mathrm{op}} = 1 / \sum_i p_i^2$ lies between 1 and M. The upper bound follows from $1 = (\sum_i p_i)^2 \leq M \sum_i p_i^2$; equality requires equal energy. Pooling all twelve

dense layers gives $M = 9,216$. Per-layer effective ranks, pooled effective rank, and numerical matrix rank have different denominators.

The global energy count $K^E_{.90} = 3,050$ is the number of dyads needed after sorting all layers' energies together. It does not mean a "top-90" budget. Uniform top-eight per layer is another selection rule: it keeps 96 dyads and 28.56% energy. A global budget and a uniform per-layer budget need not select the same components even when their total count matches.

B.4 Recovery, removal, and nonlinearity

Writing $R_0 = \mathcal{R}(\theta_0)$ and $R_A = \mathcal{R}(\theta_A)$, the actual risk corresponding to normalized removal necessity N is

$$R_{\text{remove}} = R_A + N(R_0 - R_A).$$

Sufficiency and necessity examine different anchors: $\theta_0 + P_C\Delta_s$ and $\theta_A - P_C\Delta_s$. They do not require $S_C + N_C = 1$. A nonlinear risk may depend on combinations of parameter changes, so retaining and deleting a component set are complementary tests rather than redundant measurements of one linear coefficient.

B.5 Matched Controls: What Is Held Fixed?

For a sampled dyad set J_ℓ of size k, form $R_\ell = \sum_{j \in J_\ell} \sigma_{\ell j} B_{\ell j}$ and scale it to

$$\widetilde{R}_\ell = R_\ell \frac{\| \sum_{j \le k} \sigma_{\ell j} B_{\ell j} \|_F}{\|R_\ell\|_F}.$$

Both component count and per-layer norm match the learned top set. The random set is drawn from the learned SVD dictionary; this does not make it an isotropically random subspace of the entire parameter space. Its purpose is to test the special role of the leading learned components at matched energy.

C Dense Endpoint Readout and Complete Grid

Dense endpoint readout. All twelve matrix updates have numerical rank 768 using the relative singular value threshold 10^{-6}. Pooling all 9,216 dyad energies yields $\text{ER}_{\text{op}} = 519.677$, $K^E_{.90} = 3{,}050$, and $K^E_{.95} = 4{,}416$. Uniform top-8/layer retains 28.56% of energy; top-32/layer retains 50.67%. Initial, full, keep-8, and remove-8 Granite risks are .80885, .02482, .04097, and .76942. The normalized effects are $S = .979\,[.955, 1.004]$ and $N = .950\,[.925, .972]$, using 2,000 paired-prompt bootstrap resamples. Three matched random learned-component controls give mean risk $.80197 \pm .00292$ (population standard deviation across control seeds) and mean recovery .00877. Guard 2 gives recovery .9786 and removal .9409, with random recovery .0016. These are evaluations of the same trained endpoint, not independent safety-training replications.

Table 4: Complete dense grid (k per layer).

k	1	2	4	8	16	32	64	128	256	512
Energy	.104	.147	.209	.286	.385	.507	.643	.779	.901	.981
S_k	.704	.881	.897	.979	1.001	1.008	1.008	.999	1.002	1.001
N_k	.908	.903	.927	.950	.971	.987	.991	.987	.987	1.000

D LoRA Endpoint Cores across Objectives

We compare three safety objectives under rank-32 LoRA: fixed-refusal training, harm-weighted token unlikelihood combined with refusal training, and policy-gradient minimization of the generated responses' harm scores. Geometry is measured on effective weight updates. The global budget K counts selected dyads across all layers, not per layer.

Table 5: LoRA uses a global dyad budget; S/N audited at Granite-selected $K = 12$.

Objective	$\mathrm{ER}_{\mathrm{op}}$	$K^E_{.90}$	Point/LCB K^*	Granite S/N	Guard 2 S/N
Fixed-refusal training	10.447	11	12/16	.935/.994	.937/1.003
Harm-weighted unlikelihood + refusal training	10.586	11	12/16	.908/.994	.926/.999
Harm-score minimization	9.251	10	12/12	1.003/.992	1.021/1.023

The behavioral threshold K^* is the smallest tested global budget at which both recovery and removal reach .9 and remain above .9 at every larger tested budget. We report thresholds based on point estimates and on pointwise bootstrap lower bounds. These are grid-dependent thresholds, not intrinsic ranks or simultaneous-confidence guarantees.

E Core Formation and Prefix Interventions

E.1 Current-core function and geometric change

Geometry is recorded at steps 0, 1, and every five steps thereafter; behavioral reconstructions are evaluated at the prespecified checkpoints in the table below. The diagnostics are computed after training from saved checkpoints, using only the weights available at each prefix. Each keep/remove

Table 6: Three-seed mean prefix results, T1/K4/16 held-out prompts. These risks are not pooled with the primary endpoint cohort.

Step	Actual risk	Keep risk	Remove risk	Dolly-64 NLL
0	.859072	—	—	3.138755
25	.714164	.733287	.846337	3.162249
50	.489788	.527572	.844624	3.199322
100	.256488	.322914	.836425	3.196393
150	.072020	.138192	.841603	3.196033
200	.050198	.111918	.847430	3.180195
300	.020923	.043989	.828153	3.169391
450	.010367	.038525	.828586	3.175743

row uses the SVD of that prefix's own safety update.

For the rank-96 current projector, Equation 5 defines $d_t = 1 - \mathrm{tr}(P_t P_{t-5})/96$. Equivalently, d_t is the mean squared sine of the principal angles between the two subspaces, so it is independent of the signs or ordering of their basis vectors. The diagnostic detects stabilization when $d_t \leq .05$ for five consecutive regular comparisons. The first detections are 315, 205, and 225 for seeds 340–342. Detection compares consecutive regular five-step observations, not every individual optimizer step. The first observed state after initialization is step1; its comparison to step5 has a different interval and is not by itself enough to satisfy the five-consecutive-comparison rule. Later threshold breaches occur 3, 6, and 7 times, respectively, so first detection is not permanent convergence. The threshold is a diagnostic, not a physical phase transition. SVD boundary gaps and accumulation effects also influence projector motion independently of behavioral change.

F Downstream Geometry and Interventions

F.1 Every-Step Downstream Coordinate Tracking

The three tracked downstream seeds (140–142) start from the same aligned checkpoint and use 450 AdamW steps, batch size 8, learning rate 10^{-4}, and the same twelve adapted matrices. For the downstream iterates ϑ_t and fixed vector a of Section 6, $u_t = \vartheta_t - \vartheta_{t-1}$, $q_t - q_{t-1} = \langle u_t, a\rangle / \|a\|^2$ and $\cos(u_t, a) = (q_t - q_{t-1})\|a\|/\|u_t\|$, using $P_C a = a$. Mean q_t over three tracked seeds moves from $-.00143$ at step 1 to $-.10029$ at 450, and $\cos(c_t, a)$, with $c_t = P_C(\vartheta_t - \theta_A)$, from $-.413$ to $-.865$; the three-seed mean per-step ambient cosine $\cos(u_t, a)$ ranges from approximately $-.018$ to $-.004$ and becomes *less* negative late in training, so cumulative opposition strengthens without

individual steps turning more against a. Figure 4a shows the cumulative trace; risk is available only at saved checkpoints. Per-layer contributions to the final q are uniform across the twelve layers ($-.005$ to $-.011$ each).

F.2 Multi-Task Downstream Study

Eleven tasks (Dolly brainstorming, classification, closed QA, creative writing, general QA, information extraction, open QA, summarization; HH helpful-chosen, harmless-chosen, harmless-rejected as single-turn SFT), three learning rates ($5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}$), 450 steps from θ_A, endpoints at 150 and 450 (66 models), evaluated on the 128-prompt cohort with both judges and Dolly-1000 NLL. All 66 have $q < 0$. The core share $f_C = \langle P_C\Delta_d, P_C\Delta_s\rangle / \langle \Delta_d, \Delta_s\rangle$ lies in $[.43, .75]$ (median .56). Spearman correlation of the Granite increase with $-q$ is .686, against .255 for the unsigned core energy fraction and .409 for $\|\Delta_d\|$. The analogous whole-safety-update coordinate, $-\langle \Delta_d, \Delta_s\rangle / \|\Delta_s\|^2$, has correlation .683, so this comparison does not establish predictive superiority of the core over full-update signed alignment. All conditions share one aligned model; each task/learning-rate combination has one downstream training seed, and the two saved endpoints are dependent. Figure 5 compares the three geometric measurements against the same risk increases.

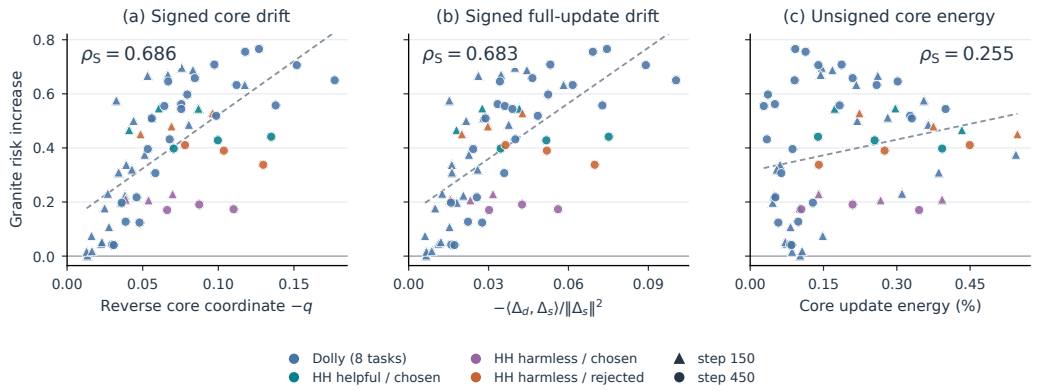


Figure 5: **Signed displacement is more associated with degradation than unsigned core energy.** Each panel uses the same 66 checkpoints from 33 task/learning-rate runs. The vertical axis is the Granite risk increase from the shared aligned model θ_A. Horizontal axes show (a) the opposing core coordinate $-q$, (b) the analogous signed coordinate along the full safety update, and (c) $100\|P_C\Delta_d\|^2/\|\Delta_d\|^2$. Colors denote task families (HH harmless +/−: chosen/rejected responses) and shapes denote checkpoints. Annotations are Spearman rank correlations; dashed lines are descriptive least-squares guides in the plotted coordinates, not rank-correlation fits or independent predictive tests.

F.3 First-Order Harm Decomposition per Interval

This probe uses 16 prompts with four full-softmax temperature-1 responses per prompt at each interval start. A prompt-wise leave-one-out score-function surrogate yields the estimate $\widehat{g}$. The associated held-out behavior uses a separate set of 16 prompts at temperature 1. This is a different study and sampling protocol from the 128-prompt temperature-.7 evaluations underlying the correlations in Figure 4b. Writing $\bar{h}_{i,-j}$ for the mean harm of the other responses to prompt i, the probe computes

$$\widehat{g} = \frac{1}{16 \cdot 4} \sum_{i=1}^{16} \sum_{j=1}^{4} (h_{ij} - \bar{h}_{i,-j}) \nabla_\theta \log p_\theta(y_{ij} \mid x_i),$$

with sampled responses and harm scores detached during differentiation. Finite differences at $\epsilon = .01, .1$ validate the same-bank surrogate derivative, not generated risk. The 27 intervals are dependent; only $0 \to 1$ is a single optimizer step. Core, complement, and total inner products are positive in 27, 22, and 24 of the 27 intervals, respectively. Through step 25, the core fraction of

Table 7: Three-seed means of estimated inner products and their core/complement terms for consecutive saved checkpoints; u is the actual interval displacement. Shares are omitted near zero or when seed-level totals change sign (–).

Interval	$\langle \widehat{g}, u \rangle$	$\langle \widehat{g}, P_C u \rangle$	$\langle \widehat{g}, (I-P_C)u \rangle$	core share (range)	$\|P_C u\|^2/\|u\|^2$	core > 0
0–1	.0105	.0009	.0096	.08–.10	.0014	3/3
1–5	.0507	.0058	.0449	.08–.13	.0020	3/3
5–10	.0776	.0091	.0685	.10–.16	.0020	3/3
10–25	.2332	.0225	.2107	.08–.13	.0014	3/3
25–50	.0286	.0229	.0057	–	.0010	3/3
50–100	.1187	.0405	.0781	.23–1.03	.0010	3/3
100–200	.0906	.0357	.0548	.33–.49	.0007	3/3
200–300	.0550	.0192	.0358	.28–.44	.0006	3/3
300–450	.0255	.0132	.0123	–	.0005	3/3

the estimated total ranges from .0788 to .1592, and the maximum seed-level core-energy fraction is .002361 (about .236%).

F.4 Post-Adaptation Editing: Design and Full Scalar Sweeps

Let $b = Q_C^\top \Delta_d$ be the downstream core coordinates and $s = Q_C^\top \Delta_s$ the safety coordinates. Then $c = Q_C b$ and $a = Q_C s$. Two edit families should not be conflated:

$$\theta_D + (\alpha - 1) Q_C b \quad \text{and} \quad \theta_D + \gamma Q_C s.$$

The first changes the observed downstream core displacement, while the second adds a safety-derived vector. Both use the same pre-downstream projector but generally different signed directions. α and γ are controlled intervention amplitudes; q is a measured reference coefficient.

Table 8: Complete registered α grid for $\theta_D + (\alpha - 1)c$: eight downstream seeds. Intervals are percentile seed-bootstrap intervals, not prompt intervals.

Variant	Granite	95% seed interval	Guard 2	Dolly NLL
$\alpha = -2$	0.0962	[0.0798,0.1147]	0.0551	2.6151
$\alpha = -1$	0.2786	[0.2511,0.3069]	0.1712	2.6127
$\alpha = -0.5$	0.4122	[0.3795,0.4432]	0.2608	2.6118
$\alpha = 0$	0.5488	[0.5286,0.5670]	0.3595	2.6111
$\alpha = 0.5$	0.6465	[0.6326,0.6588]	0.4434	2.6105
$\alpha = 1$	0.7081	[0.6962,0.7185]	0.5155	2.6100
$\alpha = 2$	0.7748	[0.7663,0.7843]	0.6039	2.6094

Table 9: Complete safety-vector boost grid, $\theta_D + \gamma a$, on the development cohort. Amplitude selection uses this grid, not the confirmation table.

Variant	Granite	95% seed interval	Guard 2	Dolly NLL
$\gamma = 0$	0.7081	[0.6962,0.7185]	0.5155	2.6100
$\gamma = 0.05$	0.6722	[0.6600,0.6835]	0.4733	2.6101
$\gamma = 0.1$	0.6303	[0.6143,0.6454]	0.4227	2.6102
$\gamma = 0.2$	0.4977	[0.4611,0.5298]	0.3204	2.6106
$\gamma = 0.4$	0.2097	[0.1785,0.2422]	0.1267	2.6120
$\gamma = 0.8$	0.0398	[0.0278,0.0533]	0.0160	2.6177

The scalar sweep is development evidence. A separate prompt confirmation of the selected safety boost is reported separately rather than being pooled with the grid.

Table 10: Separate confirmation prompt cohort. The aligned anchor is shared; its row is not five independent safety models.

Variant	Granite	95% seed interval	Guard 2	Dolly NLL
Aligned	0.0168	—	0.0242	2.9871
Downstream	0.6787	[0.6651,0.6923]	0.4848	2.6107
Selected $+.8a$	0.0254	[0.0141,0.0421]	0.0042	2.6187

Table 11: Per-seed confirmation: raw Granite/Guard 2 risk and selected safety-boost risk. The shared utility set and safety anchor remain fixed.

Seed	Granite raw	Granite boost	Guard 2 raw	Guard 2 boost	NLL cost
44	0.69680	0.05734	0.49983	0.01733	0.00718
45	0.66369	0.01743	0.46259	0.00082	0.00769
46	0.65871	0.02576	0.44243	0.00091	0.00699
47	0.69350	0.01276	0.50818	0.00074	0.00827
48	0.68070	0.01372	0.51091	0.00129	0.00978

G Qwen Replication: Functional Cores and Downstream Effects

The Qwen experiments extend three observations beyond GPT-2: functional compression of the safety update (Section 4), early current-core function during tuning (Section 5), and opposing downstream drift with useful post-hoc corrections (Section 6). Top-eight reconstructions recover almost all of the additional safety gain using about one fifth of the update energy. Removing these components raises Granite risk in both models, although the fraction of gain erased is smaller than in GPT-2: the *strength* of necessity is model-dependent. The already safe initial checkpoints make absolute risks important alongside normalized gains.

G.1 Models, training, and intervention construction

We start from Qwen2.5-0.5B-Instruct and Qwen2.5-1.5B-Instruct (Team, 2024), using each checkpoint's native chat template. These are already instruction- and safety-tuned models, not pretrained base models. Accordingly, Δ_s denotes the *additional* update learned in our run; the interventions do not isolate all safety acquired before that run. Each endpoint run in the following tables uses seed 840 and its own initialization, endpoint, and fitted projector. No projector is transferred between models.

Only the dense query, key, and value weight matrices are trained; biases and all other parameters are frozen. For spectral analysis, we concatenate the three updates along their output dimension within each transformer layer and apply the same dyad retain/remove construction as in Equation 2. Thus top eight means eight dyads of the *concatenated QKV update per layer*, not eight for each of Q, K, and V. Table 12 reports the resulting coordinate and dyad counts. Reconstruction energy closure is checked against the exact saved endpoint update.

Training uses 450 AdamW steps at learning rate 10^{-4}, $(\beta_1, \beta_2) = (.9, .999)$, $\epsilon = 10^{-8}$, zero weight decay, and gradient clipping at 1. Each step replays eight harmful prompts from the fixed curriculum, drawing four fresh full-softmax responses per prompt at temperature 1, with at most 64 new tokens. Prompt-group microbatches accumulate the leave-one-out raw-harm score-function loss before one optimizer step. The benign forward-KL anchor has weight 4 and uses fixed initial-model continuations for 128 Alpaca prompts. This is reward-based harm minimization, not supervised fixed-refusal training.

G.2 Evaluation and absolute endpoint effects

The held-out cohort contains 32 AdvBench prompts, selected from index 416 onward after excluding exact training-prompt matches. Four temperature-1 full-softmax responses are sampled per prompt, with at most 64 new tokens; Granite and Guard 2 score the same 128 generated responses for each variant. This is a separate protocol from the primary GPT-2 endpoint cohort, so raw risks are not pooled across these studies. Utility uses 64 held-out Dolly examples, with the native-chat prefix

Table 12: Qwen update geometry. Matrix shapes refer to the concatenated QKV update of one layer. $K^{E}_{.90}$ pools squared singular values across layers; E_8 is the energy retained by eight dyads per layer.

Quantity	Qwen2.5-0.5B	Qwen2.5-1.5B
Transformer layers	24	28
QKV matrix shape	1152×896	2048×1536
Trainable scalar coordinates	24,772,608	88,080,384
Available SVD dyads	21,504	43,008
Top-8/layer dyads	192	224
E_8 (%)	21.275	19.931
$K^{E}_{.90}$	5,945	10,491

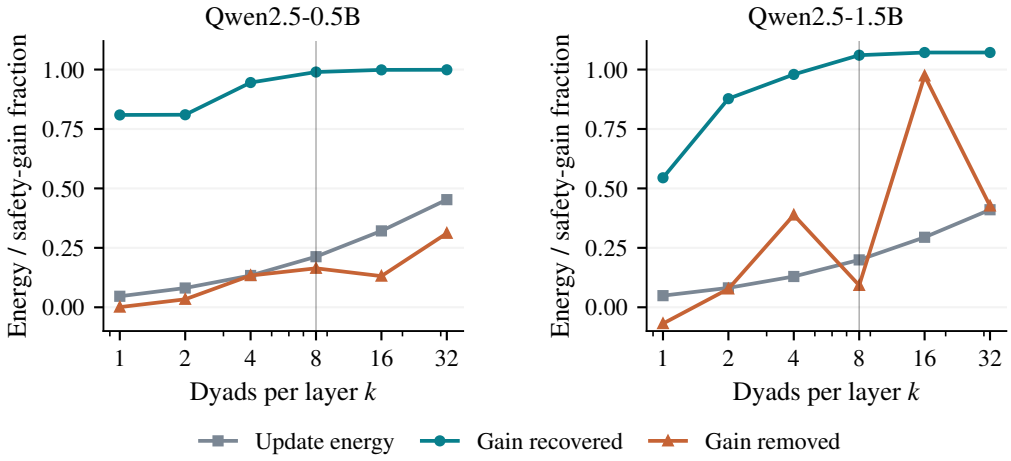


Figure 6: **Spectral energy and functional compression in Qwen.** Separate panels show the two native Instruct models, each with safety seed 840. Curves report retained energy E_k, Granite safety-gain recovery S_k, and gain erased by removal N_k over every tested per-layer dyad budget. The vertical reference marks $k = 8$. Values are point estimates; recovery above one means lower risk than the full endpoint, not greater than complete parameter retention. Compact sufficiency recurs, while removal strength differs from GPT-2.

limited to its last 256 tokens and each target to its first 255 tokens plus EOS. The evaluated target count is 4,843 tokens for each model. NLL measures response prediction within each model; it is not a shared cross-model capability scale.

Table 13 gives absolute risks and NLL, rather than only gain ratios. Keeping the top-eight contribution changes Granite risk from .147546 to .002057 for 0.5B and from .014782 to .000698 for 1.5B. Relative to each run's full safety gain, these give $S_8 = .9896$ and 1.0604. Removing the same contribution gives $N_8 = .1640$ and $.0921$: risk rises relative to the full model in both cases, while the complement retains most of the newly acquired gain. A value $S_8 > 1$ means that the reconstruction has lower measured risk than the full endpoint; it is not a fraction of model parameters or a guarantee beyond this cohort.

Table 13: Absolute Qwen endpoint readouts. Initial is the native Instruct checkpoint; full adds the entire learned safety update. Keep/remove use eight dyads per layer. Lower values are better for all three metrics.

Model	Intervention	Granite risk	Guard 2 risk	Dolly-64 NLL
0.5B	Initial	.147546	.088094	2.3895
	Full	.000525	.000213	2.4565
	Keep-8	.002057	.002303	2.3927
	Remove-8	.024629	.009554	2.4388
1.5B	Initial	.014782	.004423	2.0840
	Full	.001499	.002590	2.1788
	Keep-8	.000698	.000279	2.1113
	Remove-8	.002722	.000303	2.1244

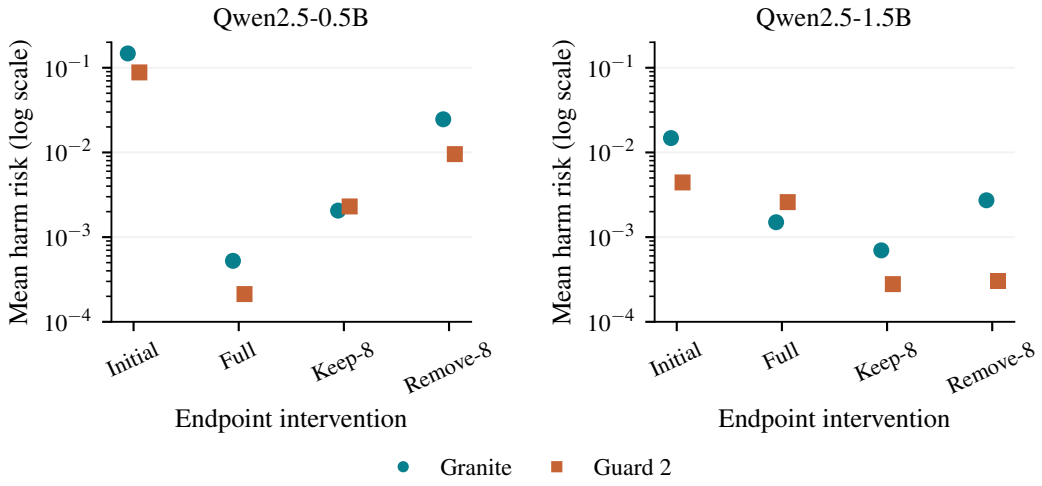


Figure 7: **Absolute Qwen endpoint risks under both judges.** Initial, full, keep-eight, and remove-eight use each model's own seed-840 safety update. The vertical axes are logarithmic. Both judges score the same 128 responses per variant; plotted values are means without uncertainty bars. On 1.5B, Guard 2 gives lower risk to removal than to the full endpoint, so the Granite removal effect is not a uniform model–judge result.

Paired-prompt bootstrap intervals use 10,000 resamples and condition on the one trained checkpoint per model. The Granite full-gain intervals are $[.08562, .21402]$ for 0.5B and $[.00561, .02687]$ for 1.5B. For Guard 2, they are $[.03545, .14917]$ and $[-.00586, .01195]$, respectively. The 1.5B Guard 2 gain is small and its interval crosses zero; moreover, its Remove-8 risk is lower than Full, unlike Granite. The removal penalty therefore replicates across the evaluated models under Granite.

G.3 DYAD BUDGET AND MODEL-DEPENDENT REMOVAL STRENGTH

Table 14 reports every tested dyad budget. The energy–behavior gap persists across the tested budgets. For Qwen2.5-1.5B, retaining 16 dyads per layer recovers the full measured safety gain, while removing them erases 97.3%, using only 29.4% of update energy. This provides a compact budget with strong paired sufficiency and necessity. The full budget sweep shows how the strength of recovery and removal varies with the selected components; it does not require a single universal cutoff to establish their functional role.

Table 14: Complete Qwen dyad-budget grid. k is per layer; E_k is the retained energy percentage; S_k, N_k use Granite and Equation 4. Values are point estimates, not confidence bounds.

	Qwen2.5-0.5B			Qwen2.5-1.5B		
k	E_k (%)	S_k	N_k	E_k (%)	S_k	N_k
1	4.631	.8090	.0007	4.887	.5447	-.0686
2	8.102	.8099	.0335	8.159	.8775	.0772
4	13.368	.9453	.1335	12.944	.9794	.3881
8	21.275	.9896	.1640	19.931	1.0604	.0921
16	32.125	.9986	.1310	29.446	1.0716	.9733
32	45.274	.9990	.3112	41.026	1.0717	.4262

Three random dyad-space projections per model give Granite risks .148223–.154095 for 0.5B and .014706–.021168 for 1.5B. These controls project the update onto independently sampled rank-eight dyad spaces; they match subspace dimension but *not* retained update energy.

We report the directly comparable energy fraction and $K^E_{.90}$ rather than combine effective-rank values: the Qwen analysis records an entropy-based effective rank, whereas the primary GPT-2 analysis uses an energy participation ratio. The additional experiments below measure current-core function and downstream effects separately.

G.4 CURRENT-CORE FUNCTION DURING A SAFETY-TUNING RUN

We repeat safety tuning with seed 850 and construct each tested checkpoint's core from its own cumulative safety update, without using a future basis. The model, training recipe, and 32-prompt evaluation design are unchanged. These are separate runs from the seed-840 endpoints above, not interventions that transfer a projector between runs. Already at step 25, the 0.5B current core recovers 94.87% of the measured gain, increasing to 98.99% at step 50. For 1.5B, step 50 recovery is 92.46%; at step 25 the Granite gain interval includes zero, so we do not interpret a normalized recovery ratio. Figure 8 uses the same two measurements as the main formation figure: current-core keep/remove risk, and continuous projector overlap. Functional concentration is already present while the current basis remains far from the final basis. At step 50, overlap with the final core is only .0623 and .0684 for 0.5B and 1.5B, respectively, despite the 98.99% and 92.46% recovery above.

Adjacent-core overlap increases from .430/.519 at step 10 to .923/.935 at step 450. Over the eleven recorded checkpoints from steps 400 to 450, mean overlap is .903/.935, with ranges [.884,.931]/[.917,.942], respectively. These high and comparatively steady late-training overlaps support substantial local geometric stabilization after functional concentration has already emerged. High and comparatively steady late-training overlap indicates local geometric stabilization, while allowing residual drift over longer training intervals.

At the final checkpoint, roughly one fifth of the update energy again recovers almost the entire safety gain. Removing the core raises Granite risk relative to both the full and retained-core models: for 0.5B, the remove-only risk is approximately six times the retain-only risk and eighteen times the full-model risk. For 1.5B the corresponding factors are about 1.6 and 2.3, with much smaller absolute differences. Thus, compact sufficient reconstructions and a removal penalty recur in a second safety run for each model. Together with the overlap trajectories, these measurements show early functional concentration followed by substantial local geometric stabilization.

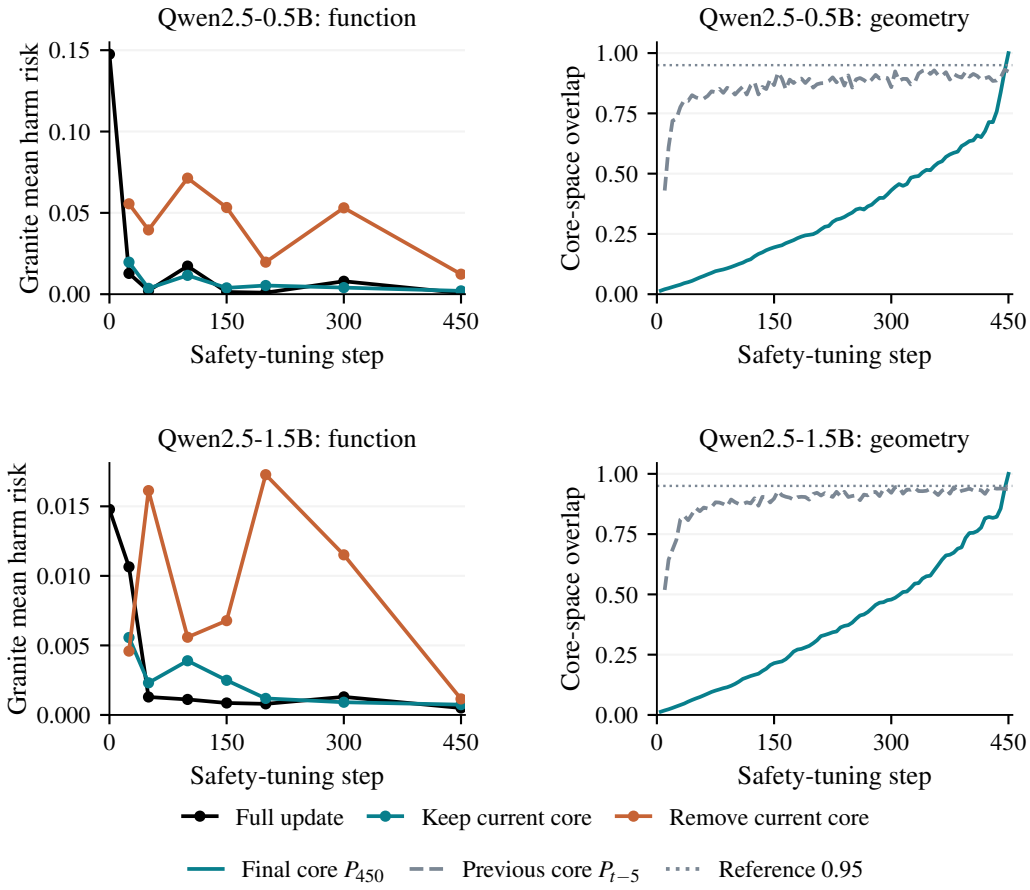


Figure 8: **Early functional concentration and increasing geometric stability in Qwen.** Top row: Qwen2.5-0.5B; bottom row: Qwen2.5-1.5B, each safety seed 850. Left: held-out Granite risk for the full update, current-core retention, and current-core removal at the seven evaluated checkpoints. The full-update curve also includes the initial checkpoint; lines only connect measured values. Right: normalized overlap $\mathrm{tr}(P_t P_s)/\dim(C)$ with the previous recorded core ($s = t - 5$) and final core ($s = 450$). Geometry is recorded every five steps; the final-core comparison is retrospective, whereas each intervention uses only its own checkpoint's cumulative update. The .95 reference is the overlap corresponding to the main study's operational distance threshold, not a requirement for functional compression. Each row is one training run; no across-seed uncertainty band is implied.

Table 15: Fresh-run (seed 850) endpoint current-core interventions. Keep/remove use eight dyads per layer; S_8, N_8 are Granite point estimates. These results are distinct from the seed-840 endpoint grid above.

Quantity	Qwen2.5-0.5B	Qwen2.5-1.5B
Core energy (%)	21.746	18.534
$K^E_{.90}$	5,841	10,885
Granite initial	.147546	.014782
Granite full	.00070	.00050
Granite keep	.00204	.00074
Granite remove	.01231	.00115
Guard 2 full	.00025	.00022
Guard 2 keep	.00029	.00022
Guard 2 remove	.00854	.00024
S_8	.9909	.9833
N_8	.0790	.0455

G.5 Downstream signed drift and endpoint corrections

For each model, a separate Dolly run (downstream seed 140) begins at its own seed-840 safety endpoint and uses that endpoint's fixed projector. It does not begin at the fresh seed-850 endpoint in Table 15. We use 450 downstream steps and the same held-out risk and Dolly-64 NLL readouts as above. Defining $a = P_C\Delta_s$ and $c = P_C\Delta_d$ as in Section 6, the final geometry is:

Table 16: Qwen downstream endpoint geometry. Energy shares are percentages; q is a signed coefficient, not an energy fraction.

Quantity	Qwen2.5-0.5B	Qwen2.5-1.5B
$100\|c\|^2/\|\Delta_d\|^2$	.17025	.05431
$\cos(c, a)$	−.61849	−.84284
$q = \langle c, a\rangle/\|a\|^2$	−.07564	−.06123
$\cos(\Delta_d, a)$	−.02552	−.01964

Figure 9 plots the complete recorded traces. Both models reproduce the main signed-drift pattern: the cumulative coordinate remains negative throughout all 450 downstream steps, and a small-energy in-core displacement accumulates opposite to the learned safety direction. Individual increments are negative at 410 and 436 steps, respectively.

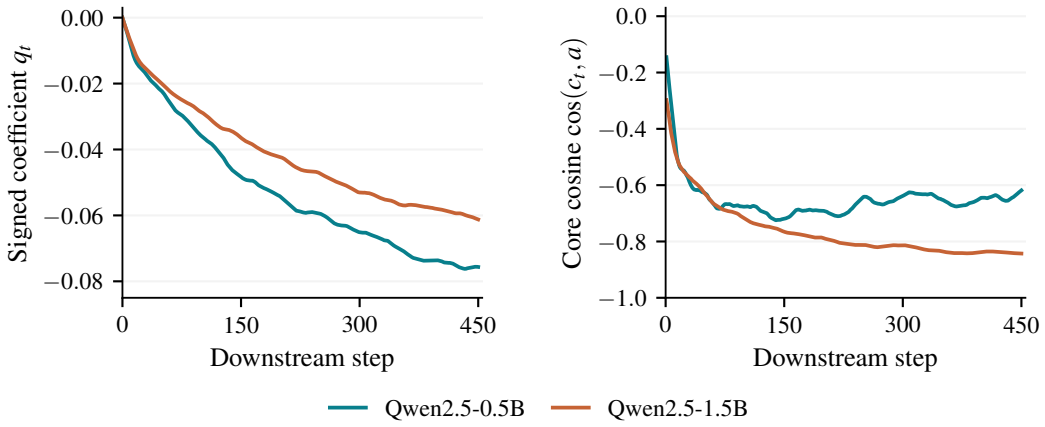


Figure 9: **Qwen downstream displacement opposes its learned safety direction.** All 450 recorded downstream steps are shown for one run per model, starting from its own seed-840 safety endpoint with its fixed top-eight/layer core. The two panels show $q_t = \langle c_t, a\rangle/\|a\|^2$ and $\cos(c_t, a)$; these are cumulative displacements, not individual gradients or current-risk derivatives. At initialization $q_0 = 0$, while the undefined cosine of zero displacement is omitted.

Figure 10 shows the same edits as safety–utility trade-offs. The boost retains 82.24% and 88.98% of ordinary fine-tuning's Dolly NLL improvement while lowering risk under both judges. Removing or reversing the downstream core displacement lowers risk under both judges in both models. Adding the safety-derived direction with the GPT-2-selected coefficient, $\theta_D + .8a$, further reduces risk while retaining 82.24% and 88.98% of the ordinary fine-tuning Dolly NLL improvement. These results extend both signed opposition and post-hoc recoverability beyond GPT-2.

H Llama-2-7B: Functional Cores and Downstream Effects

We extend the endpoint interventions, current-prefix measurements, and downstream signed edits to Llama-2-7B-Chat. The endpoint and downstream experiments share safety seed 840; a separate seed-850 run measures formation. Each projector is fitted within its own safety run. These experiments concern Sections 4–6, not the online correction method.

Table 17: Qwen downstream endpoint interventions. Remove uses $\theta_D - c$, reverse uses $\theta_D - 2c$, and boost uses $\theta_D + .8a$. The boost coefficient is carried over from the GPT-2 study, not selected on these Qwen results.

Model	Variant	Granite	Guard 2	Dolly NLL
0.5B	Aligned	.00052	.00021	2.45652
	Ordinary FT	.70402	.52650	2.30452
	Remove	.57788	.43224	2.30564
	Reverse	.53446	.42226	2.30864
	Boost	.21656	.13349	2.33152
1.5B	Aligned	.00150	.00259	2.17880
	Ordinary FT	.27867	.21815	1.99427
	Remove	.21176	.14462	1.99488
	Reverse	.23140	.18092	1.99583
	Boost	.03413	.01622	2.01460

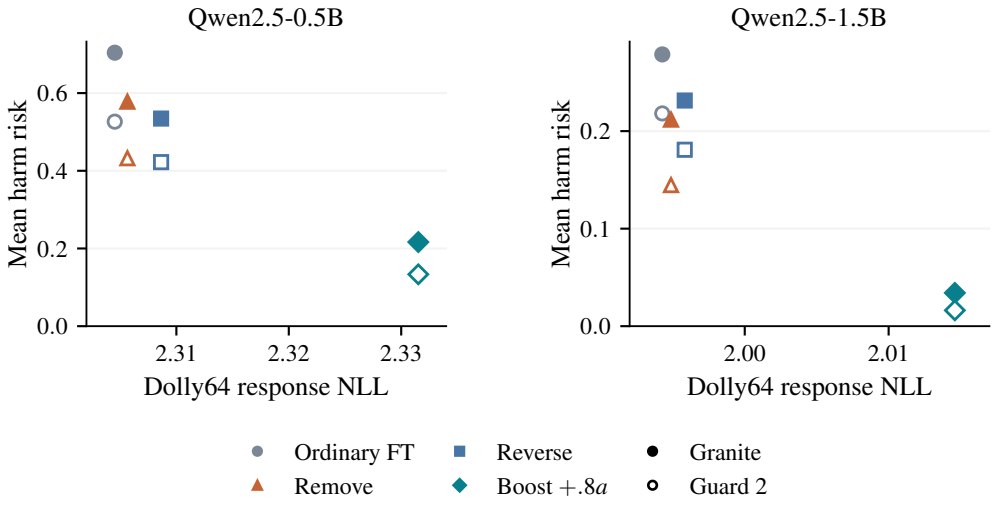


Figure 10: **Post-hoc core edits after Qwen downstream tuning.** Each panel reports ordinary FT θ_D, removal $\theta_D - c$, reversal $\theta_D - 2c$, and the fixed safety boost $\theta_D + .8a$. Filled/open markers denote Granite/Guard 2 on the same response banks. Lower risk and lower Dolly-64 response NLL are better.

H.1 ENDPOINT FUNCTIONAL CONCENTRATION

Llama-2-7B exhibits the same separation between spectral energy and behavioral importance. Keeping 64 dyads per layer recovers 94.2% of the safety gain using 36.5% of update energy, while removing them erases 73.6%. At 128 dyads per layer, recovery and removal increase to 97.2% and 81.1%, respectively, using 49.2% of the energy. Both interventions identify a compact, behavior-carrying component of the safety update.

Protocol. We start from `meta-llama/Llama-2-7b-chat-hf` (Touvron et al., 2023) and use its native chat template, with seed 840. This is additional safety tuning of an already aligned model, not base-model RLHF. As in the Qwen endpoint study, we train only dense QKV weights for 450 AdamW steps at learning rate 10^{-4}, with zero weight decay and clipping at 1. The raw-harm policy-gradient objective uses eight curriculum prompts per step, four full-softmax samples per prompt, temperature 1, and up to 64 new tokens. A forward-KL anchor of weight 4 uses frozen initial-model continuations for 128 Alpaca prompts. Forward computation uses BF16, with FP32 trained QKV weights and optimizer states. Each of the 32 layers contributes one concatenated 12288×4096 QKV update. There are 131,072 available dyads, of which 34,944 are required for 90% of the energy.

Evaluation uses 32 held-out AdvBench prompts, excluding exact training matches, with four temperature-1 responses of at most 64 tokens per prompt. Both judges score the same generated responses. The table reports the complete tested dyad grid.; $k = 128$ is its largest budget, not an estimated minimal compression point.

Table 18: Llama-2-7B endpoint interventions. k is per layer; E_k is the energy fraction, S_k recovery, and N_k removal effect. Ratios are Granite point estimates for the additional gain in this run.

k	E_k (%)	Keep risk	Remove risk	S_k	N_k
1	3.65	.014541	.000484	-2.543	.022
2	5.46	.009952	.000406	-1.394	.002
4	8.13	.002709	.002247	.421	.464
8	12.14	.002645	.001841	.437	.362
16	17.93	.001781	.001645	.653	.313
32	25.95	.001109	.001793	.822	.350
64	36.46	.000630	.003335	.942	.736
128	49.21	.000509	.003636	.972	.811

Absolute effects and scope. Initial and full Granite risks are .004389 and .000397. At $k = 128$, retaining the core gives risk .000509, recovering 97.2% of the safety gain with 49.2% of update energy. Removing the same components raises risk to .003636, erasing 81.1% of the gain. The corresponding absolute risk reduction from initialization and increase from the full endpoint are .003880 $[.002851, .005131]$ and .003239 $[.002484, .004134]$, respectively. Brackets denote paired-prompt 95% bootstrap intervals from 10,000 resamples, conditional on the trained safety checkpoint. Together, these interventions identify a behavior-carrying core through both recovery and removal, consistent with the energy–behavior separation observed in GPT-2 and Qwen.

Guard 2 initial/full risks are .000302/.000256, and keep-128/remove-128 risks are .000286/.000270. Because the initial-to-full gain is below 10^{-4}, we report absolute risks rather than normalized effects for this judge. Dolly-64 response NLL is 2.2246 initially and 2.6033 after full tuning; keep-128/remove-128 give 2.5599/2.1821. These interventions measure the safety gain acquired during the analyzed tuning run relative to the initial Chat checkpoint.

H.2 CURRENT-CORE FUNCTION DURING SAFETY TUNING

We examine when functional concentration emerges by analyzing the safety-tuning trajectory with seed 850 at steps 25, 50, 100, 150, 300, and 450. At each checkpoint, we compute the SVD of the cumulative update $\Delta_t = \theta_t - \theta_0$ and construct its own top-k/layer projector $P_{t,k}$ for $k = 8$ and 32, giving core dimensions of 256 and 1,024. The keep/remove models are $\theta_0 + P_{t,k}\Delta_t$ and $\theta_t - P_{t,k}\Delta_t$, respectively. Each intervention uses only the update available at that checkpoint, allowing us to assess current-core function independently of agreement with the final core.

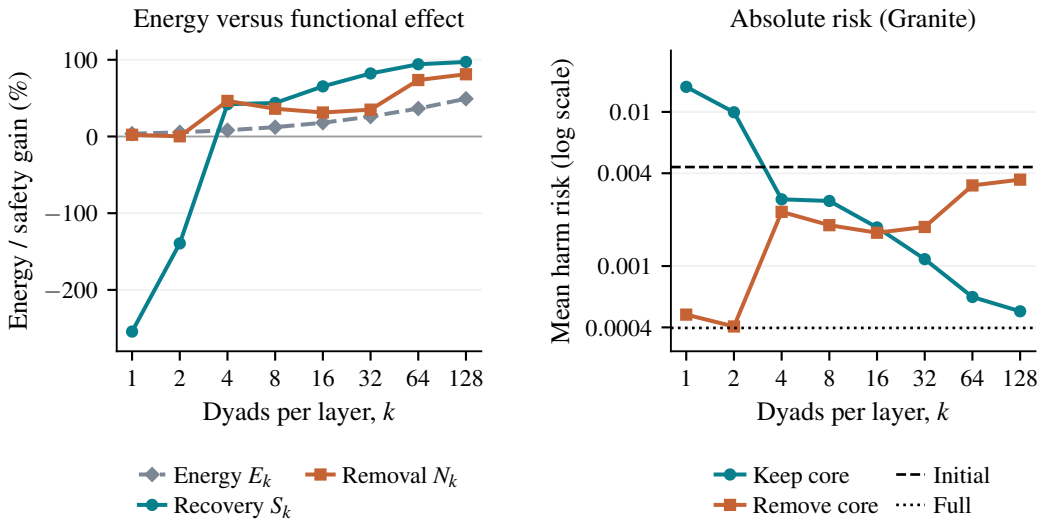


Figure 11: **Llama-2-7B endpoint functional concentration.** Left: retained energy, safety-gain recovery, and gain erased by removal across all evaluated per-layer budgets. Right: absolute Granite risk with initial and full checkpoints as references.

Table 19: Llama current-prefix interventions, safety seed 850. Entries are Granite mean risk on the same 32-prompt evaluation cohort. The initial risk is .004389. Both prespecified budgets and every evaluated prefix are shown.

Step	Full	Keep-8	Remove-8	Keep-32	Remove-32
25	.004609	.015164	.003596	.018315	.003509
50	.002694	.003630	.003233	.002879	.002914
100	.003278	.003882	.002884	.003606	.003432
150	.000900	.003188	.008618	.002250	.002941
300	.001784	.006152	.047893	.001425	.002350
450	.014024	.006440	.053851	.028956	.002072

At step 50, the top-32 core carries 30.14% of update energy and recovers 89.1% of the measured safety gain; removing it erases 13.0%. At step 150 these effects are 61.3% and 58.5%, with 27.86% energy. At step 300, 26.58% energy recovers 113.8% of the gain, while removal erases 21.7%. Recovery above one means the reconstruction is safer than the full checkpoint, not a probability above one. These current-prefix interventions show that functional concentration emerges before training is complete and while the core geometry is still evolving.

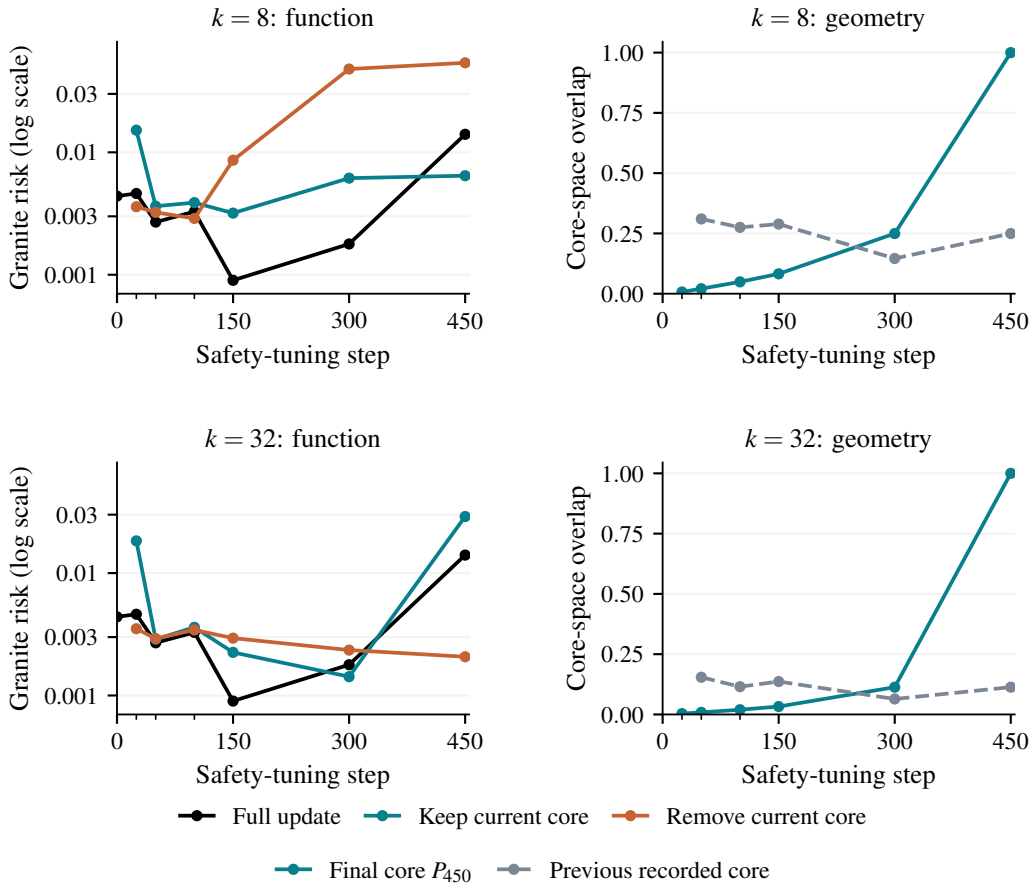


Figure 12: **Llama current-core function and evolving geometry.** Top row: $k = 8$ per layer; bottom row: $k = 32$, both from the fresh seed-850 run. Left: full, current-core keep, and current-core remove risk on a logarithmic axis, including every evaluated prefix. Right: normalized projector overlap with the final core and the previous recorded core. The previous-checkpoint lags range from 25 to 150 steps; these are not five-step comparisons. Final-core overlap is retrospective, while every keep/remove construction uses only its own prefix.

Function and geometry are distinct measurements. For each budget, we report $O_{t,s} = \mathrm{tr}(P_{t,k}P_{s,k})/(32k)$. The step-50 top-32 core has overlap only .0086 with the final core despite its 89.1% recovery; the corresponding overlaps at steps 150 and 300 are .0328 and .1132. Thus useful current directions need not already agree with the eventual endpoint basis. Because the geometry is sampled sparsely and at unequal intervals, this experiment does not estimate the five-step stabilization time in Equation 5.

Gain qualification and the final checkpoint. Normalized effects require a mean gain above 10^{-4} and a conditional paired-prompt 95% bootstrap interval wholly above zero. Granite satisfies this gate at steps 50, 150, and 300, but not 25, 100, or 450; Guard 2 does not satisfy it at any measured prefix.

At step 450, the Granite point estimate is .014024 versus .004389 initially, but its gain interval is $[-.03092, .00454]$. Two of 128 responses account for 96.4% of the final Granite risk.

A prefix from the original safety run. We also evaluate the already saved step-150 checkpoint of seed 840, separately from the seed-850 trajectory. Its Granite risk is .001217. Its own top-8 core carries 14.82% energy, with keep/remove risks .002908/.002180 (46.7% recovery and 30.4% removal). At $k = 32$, the corresponding risks are .003333/.003223 (33.3% and 63.3%), with 28.66% energy. Neither result transfers a projector between runs. Guard 2 does not resolve a positive full-prefix gain; its keep-32 risk is .015684. This additional prefix supports functional contributions before the endpoint, without implying that a larger budget is always behaviorally better.

H.3 Downstream signed drift and post-hoc corrections

The downstream study starts from the original seed-840 safety endpoint, not from the fresh seed-850 run above. We keep its top-eight/layer projector P_C fixed and fine-tune on Dolly for 450 steps with downstream seed 140, batch eight, AdamW learning rate 10^{-4}, zero weight decay, and clipping at one. Only dense QKV weights are updated. The actor uses single-GPU eager attention with BF16 forward computation and FP32 QKV parameters and Adam states. All diagnostic interventions below are evaluated with SDPA using the same native-chat, 32-prompt/four-response protocol and Dolly-64 examples.

Writing $a = P_C\Delta_s$ and $c_t = P_C(\vartheta_t - \theta_A)$, the endpoint geometry is:

Table 20: Llama downstream endpoint geometry. The core is the fixed 256-dimensional safety-derived dyad space. The coefficient q and signed inner-product share f_C are not energy fractions.

Quantity	Value
Core energy $100\|c\|^2/\|\Delta_d\|^2$	.04356%
Core cosine $\cos(c, a)$	$-.78010$
Signed coordinate $q = \langle c, a\rangle/\|a\|^2$	$-.05088$
Whole-update cosine $\cos(\Delta_d, a)$	$-.01628$
Signed opposition share $f_C = \langle c, a\rangle/\langle\Delta_d, \Delta_s\rangle$	.27532

The small core component again points against the learned safety direction. Summing Frobenius inner products over the adapted matrices gives $\langle c, a\rangle = -35.58146$ and $\langle\Delta_d, \Delta_s\rangle = -129.23648$. Thus 0.04356% of downstream energy carries 27.53% of the signed geometric opposition, not 27.53% of the observed risk increase. The cumulative q_t is negative at all 450 recorded steps, with 420 negative individual increments. Its trajectory is not strictly monotone; the full unsmoothed trace is shown in Figure 13.

Table 21: Llama post-adaptation edits, one downstream run from its own seed-840 safety endpoint. All rows use the same diagnostic evaluation. The coefficient .8 is carried over from GPT-2, not selected on Llama.

Model / correction	Granite	Guard 2	Dolly NLL
Aligned θ_A	.000407	.000252	2.60306
Ordinary FT θ_D	.347003	.241787	1.74468
Remove $\theta_D - c$	.297594	.197708	1.75077
Reverse $\theta_D - 2c$	.289851	.186983	1.75547
Boost $\theta_D + .8a$	.115795	.081399	1.83895

The fixed safety boost reduces Granite and Guard 2 risk by 66.6% and 66.3% relative to ordinary fine-tuning while preserving 89.0% of its Dolly NLL improvement, computed as $(\mathrm{NLL}_A - \mathrm{NLL}_{\mathrm{boost}})/(\mathrm{NLL}_A - \mathrm{NLL}_D)$. It does not fully restore the near-floor aligned risk. Removal and reversal also improve both point estimates, with 99.3% and 98.7% of the NLL gain retained, respectively.

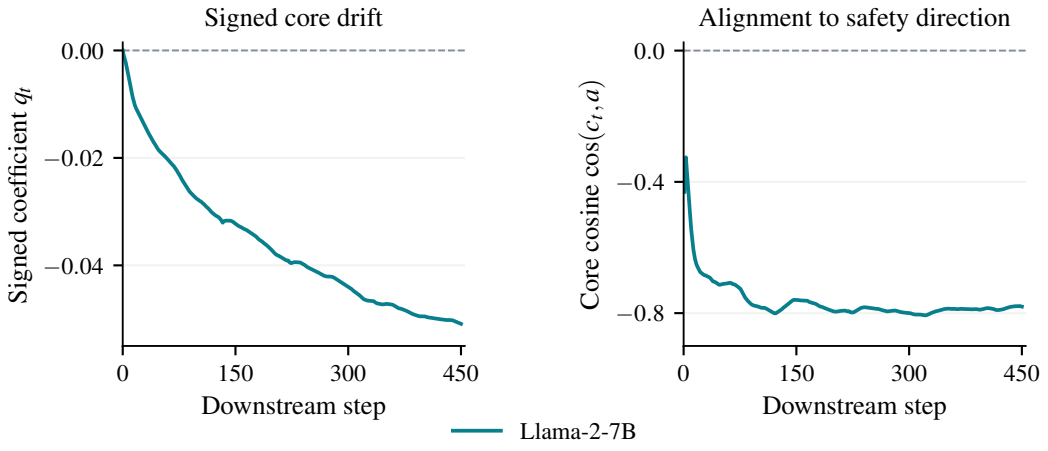


Figure 13: **Signed downstream drift in Llama.** Left: the cumulative coefficient $q_t = \langle c_t, a \rangle / \|a\|^2$. Right: $\cos(c_t, a)$, using the same fixed top-eight/layer safety core. All 450 downstream updates are shown; the undefined zero-displacement cosine at initialization is omitted. These are historical displacement coordinates, not current-risk-gradient measurements.

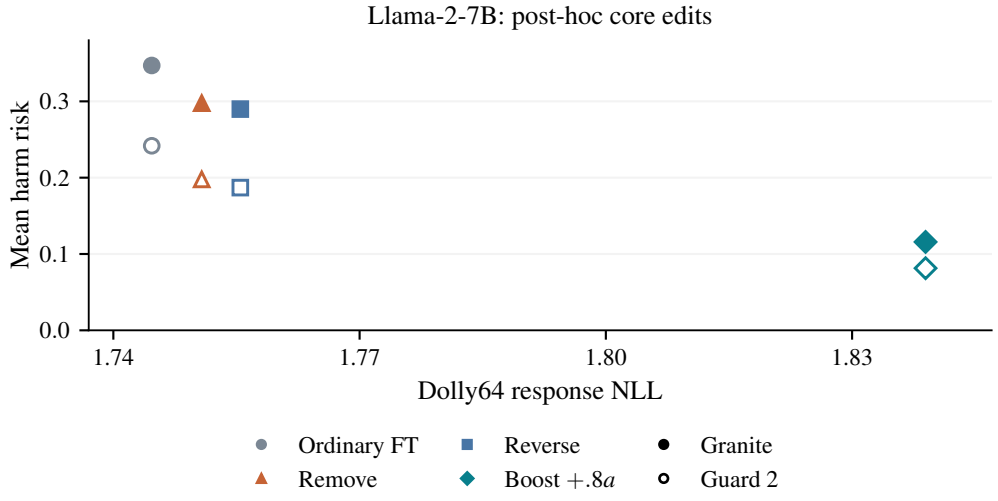


Figure 14: **Llama post-hoc safety–utility trade-offs.** The points are the four downstream endpoints in Table 21; lower risk and lower NLL are better. Filled/open markers denote Granite/Guard 2 on the same response banks. The aligned checkpoint, whose NLL is 2.60306, is reported in the table rather than included in this downstream-scale plot.

An additional descriptive paired-prompt bootstrap (20,000 resamples) gives boost risk reductions of .23121 $[.17452, .29266]$ for Granite and .16039 $[.10334, .22393]$ for Guard 2. The corresponding removal/reversal intervals include zero. These intervals condition on one trained endpoint and its 32-prompt response banks, not training-seed uncertainty; they are post-analysis uncertainty estimates rather than registered selection tests. All edits produce nonempty responses. Together, the signed measurements and finite interventions extend the evidence for post-adaptation actionability of the safety core, without establishing opposition as the sole cause of degradation. No current-harm-gradient probe from Section 6 or online correction experiment is included in these Llama diagnostics.

Evaluation provenance. The seed-140 ordinary endpoint here is bit-identical to the separately evaluated seed-140 Vanilla checkpoint in the online-method study. Diagnostic evaluation uses SDPA, whereas that comparison uses eager attention; generated responses and NLL therefore differ despite identical weights and evaluation seeds. The values must not be interchanged across studies. Every comparison within the present tables uses the same evaluation implementation.

I Downstream Correction: Controls and Comparison Protocol

The application and its initialization. Our application is the downstream update correction in Equation 9, not a replacement safety-alignment objective. In the primary experiment, its starting checkpoint and projector are the original harm-policy-gradient safety endpoint and its top-eight dyads per QKV matrix. Neither is refitted using downstream test outcomes. The memory consists of the original 64 harmful training prompts and the target "I'm sorry, but I can't help with that." Four memory examples are used per step. The projector is fixed, the memory gradient is recomputed at the current model, and the correction norm is capped at the task proposal norm. AdamW moments contain task gradients only.

For an orthonormal core basis B, write $z = Bb$. The Lagrangian of the linearized constraint is $\frac{1}{2}\|b\|^2 + \lambda[h_t^\top u_t + (B^\top h_t)^\top b]$. Stationarity gives $b = -\lambda B^\top h_t$; complementary slackness gives $\lambda = [h_t^\top u_t]_+ / \|B^\top h_t\|^2$ whenever this denominator is positive. The norm cap is applied after this unconstrained minimum-norm solution and can leave a positive linearized residual.

Correction-space ablation. Table 22 changes only the correction space while retaining the memory loss, task batches, optimizer, and correction cap. Its comparisons address the role of the correction space, whereas the main comparison evaluates different protection procedures. All three downstream seeds share one safety checkpoint and one discovered core; they are not three independent discoveries.

Correction space	Granite ↓	Guard 2 ↓	Dolly NLL ↓
None	.7150	.5266	2.6105
$\mathrm{span}(a)$	.6705	.4801	2.6110
Full QKV	.6857	.4961	2.6107
Random rank-96	.7168	.5234	2.6146
Learned core C	**.2569**	**.1612**	2.6129

Table 22: Original correction-space ablation, averaged over three downstream seeds. The starting model has Granite risk .0133 and Dolly NLL 2.9871. The corrected arms share the memory loss and norm cap, but their realized correction norms are not matched.

Common downstream protocol. The primary comparison uses `vicgalle/gpt2-alpaca`, and updates only its twelve dense attention QKV weight matrices. All trained comparison arms use the same Dolly examples and task-batch ordering for seeds 140, 141, and 142: 450 updates, task batch size eight, maximum length 512, learning rate 10^{-4}, AdamW with betas $(.9, .999)$, epsilon 10^{-8}, no weight decay, and gradient clipping at one. Training uses FP32 and the original dropout convention. SafeLoRA acts after this training rather than adding task updates. Evaluation uses the same 128 harmful prompts, two responses per prompt, temperature .7, top-k 50, top-p .92, and 64 generated tokens. Granite and Guard 2 judge the same responses; Dolly utility uses the same 1,000 held-out examples. The evaluation cohort has been used in earlier experiments, so this is not a new untouched confirmation set.

Weight-space and instruction baselines. SafeLoRA applies the authors' alignment operator $DD^\top/\|D\|_F$ to selected downstream weight updates in output-coordinate convention; it is not replaced with an orthogonal projector. The two fixed cosine thresholds .5 and .9 produce the same aggregate results here. LoX extrapolates before downstream training by adding the rank-k truncated alignment difference with coefficient one. We report both fixed budgets $k = 8$ and $k = 32$, rather than choosing a test-set winner. These are weight-space implementations on the selected QKV matrices, not reproductions of the original papers' model and benchmark settings. SafeInstr retains the original policy-gradient-aligned start and mixes safety-response examples from the sealed alignment training pool: at step t, the cumulative extra example count is $\lfloor 8t(.025/.975) \rfloor$, yielding 92 examples over 450 steps. Cross-entropy is response-token averaged over the combined batch.

Alignment-stage baselines. Vaccine and Booster perform supervised safety-response training with their respective perturbation mechanisms. Their alignment checkpoints use 5,000 safety-response examples, 10,000 updates, batch size ten, learning rate 5×10^{-4}, weight decay .1, and rank-32 LoRA with scaling parameter $\alpha = 4$. Vaccine uses hidden-state perturbations with radius two. Booster uses perturbation magnitude .1 and coefficient five, with additional harmful-response supervision. The adapters are merged before the common dense downstream training. Ordinary supervised safety tuning without either protection provides their reference. These methods therefore start downstream training from separately trained checkpoints; alignment data, budgets, and auxiliary supervision are method-specific.

Method	Before-FT NLL	After-FT NLL	After-FT Granite
Vanilla	2.9871	2.6105	$.7150 \pm .0147$
SafeLoRA (.5/.9)	2.9871	2.9728	$.0206 \pm .00002$
LoX ($k = 8$)	3.1347	2.6141	$.4096 \pm .0769$
LoX ($k = 32$)	3.1318	2.6166	$.3734 \pm .0675$
SafeInstr	2.9871	2.6086	$.6134 \pm .0398$
Ours	2.9871	2.6129	$.2569 \pm .0684$
Safety SFT (no protection)	4.0465	2.6548	$.8603 \pm .0022$
Vaccine	3.7975	2.6648	$.8496 \pm .0077$
Booster	2.8719	2.6377	$.8542 \pm .0077$

Table 23: Starting utility and downstream-seed variability for the main comparison. Risk values are means and sample standard deviations over three seeds, not confidence intervals. Before-FT LoX metrics include extrapolation; after-FT SafeLoRA metrics include its post-hoc intervention.

I.1 Model extensions: Qwen2.5-0.5B/1.5B-Instruct

We apply the same correction to each model's own completed safety endpoint and top-eight dyads per concatenated QKV block (192/224 core dimensions). The GPT-2 projector is not transferred. We retain the same 64 memory prompts, fixed refusal target, four memory examples per step, norm cap one, dense-QKV parameter updates, task-only AdamW moments, learning rate 10^{-4}, batch eight, and 450 Dolly updates. Native chat formatting replaces the GPT-2 prompt template. Both arms use deterministic eager attention; a two-step probe-on/off check gives exactly identical uncorrected weights. Evaluation follows the model-extension protocol: 32 held-out prompts with four responses each, temperature one and 64 generated tokens, and 64 Dolly validation examples. These absolute NLLs and risks should not be pooled with the different GPT-2 evaluation cohort.

These Vanilla baselines are newly trained with deterministic eager attention, rather than reused from the SDPA signed-drift and post-hoc study in Appendix G.5. Vanilla and core correction share the execution protocol within the present comparison.

Both judges improve in all three paired runs. Mean Granite risk decreases by 46.3% and Guard 2 risk by 54.0%, with a mean Dolly NLL increase of .00058. This extends the observed correction benefit beyond GPT-2 without asserting model-independent protection.

On Qwen2.5-1.5B, Guard 2 improves in all three paired runs, but Granite worsens in two; utility is nearly unchanged. This does not establish model-independent behavioral protection. The correction is active in 54.0% of steps and its mean norm is 4.00% of the task displacement norm; the cap never

Seed	Method	Granite ↓	Guard 2 ↓	Dolly NLL ↓
140	Vanilla	.6710	.5089	2.30409
140	Core correction	.2962	.1832	2.30548
141	Vanilla	.6887	.4843	2.29482
141	Core correction	.4103	.2750	2.29494
142	Vanilla	.6997	.4912	2.30271
142	Core correction	.4002	.2247	2.30293
Mean	Vanilla	.6865	.4948	2.30054
Mean	Core correction	**.3689**	**.2276**	2.30112

Table 24: Completed Qwen2.5-0.5B method extension. Three downstream seeds share the model's own safety checkpoint and core; this is not a prior-method comparison on Qwen or three independent safety-training runs.

Seed	Method	Granite ↓	Guard 2 ↓	Dolly NLL ↓
140	Vanilla	.2939	.2259	1.99419
140	Core correction	.3250	.1843	1.99578
141	Vanilla	.3851	.2928	1.99337
141	Core correction	.4405	.2869	1.99178
142	Vanilla	.4237	.2751	1.98786
142	Core correction	.4018	.2606	1.98704
Mean	Vanilla	.3676	.2646	1.99181
Mean	Core correction	.3891	.2439	1.99153

Table 25: Completed Qwen2.5-1.5B method extension under the same extension protocol. The two judges do not show a consistent aggregate improvement.

binds. For comparison, these values are 58.7% and 4.18% on 0.5B, also without cap activation. In both models, the uncapped linearized memory-loss constraint is satisfied to numerical precision. Its satisfaction therefore need not imply lower independent judge risk: it constrains a local fixed-refusal surrogate, not the behavioral evaluation objective.