Safety Subspaces are Not Distinct: A Fine-Tuning Case Study

Kaustubh Ponkshe[1,*], Shaan Shah[2,*], Raghav Singhal[1,*], Praneeth Vepakomma[1,3]
[1]Mohamed bin Zayed University of Artificial Intelligence,
[2]University of California San Diego, [3]Massachusetts Institute of Technology

* denotes equal contributions. Author order determined randomly.

Abstract

LLMs rely on safety alignment to produce socially acceptable responses. This is typically achieved through instruction tuning and reinforcement learning from human feedback. However, this alignment is known to be brittle: further fine-tuning, even on benign or lightly contaminated data, can degrade safety and reintroduce harmful behaviors. A growing body of work suggests that alignment may correspond to identifiable geometric directions in weight space, forming subspaces that could, in principle, be isolated or preserved to defend against misalignment. In this work, we conduct a comprehensive empirical study of this geometric perspective. We examine whether safety-relevant behavior is concentrated in specific subspaces, whether it can be separated from general-purpose learning, and whether harmfulness arises from distinguishable patterns in internal representations. Across both parameter and activation space, our findings are consistent: subspaces that amplify safe behaviors also amplify unsafe ones, and prompts with different safety implications activate overlapping representations. We find no evidence of a subspace that selectively governs safety. These results challenge the assumption that alignment is geometrically localized. Rather than residing in distinct directions, safety appears to emerge from entangled, high-impact components of the model's broader learning dynamics. This suggests that subspace-based defenses may face fundamental limitations and underscores the need for alternative strategies to preserve alignment. We corroborate these findings through multiple experiments on five open-source LLMs. Our code is publicly available at: https://github.com/CERT-Lab/safety-subspaces.

1 Introduction

Large Language Models (LLMs) have shown strong performance across a wide range of general-purpose tasks, including complex reasoning and problem solving (1; 43; 49–51; 58). To ensure these models behave responsibly and align with human values, they undergo an additional process of *security alignment*. Despite known jailbreak methods that can bypass safeguards, aligned models are generally considered significantly safer than their base versions (37; 44; 55). This alignment is typically achieved through supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF), enabling models only improve response quality, and also refuse harmful or inappropriate prompts. A growing line of research focuses on the weight difference between the base and aligned models, commonly referred to as the *alignment matrix*, which captures the transition from unaligned to aligned behavior. This difference has been used to interpret alignment mechanisms and develop defenses against adversarial attacks (3; 10; 18; 25; 28; 30; 57).

However, this alignment is fragile. Since safety is encoded in the model's weights, any modification, such as further fine-tuning (FT), can compromise it. While FT adapts models to new tasks by learning update directions, it offers no guarantee that safety is preserved. This exposes a deeper attack surface beyond prompt engineering: an adversary could insert a small number of malicious samples into a training set to subvert alignment (4; 59; 60; 64). Recent work shows that even benign FT, low-rank adaptation, or pruning can degrade a model's safety profile (15; 16; 29; 42; 54). Preserving alignment under continued training is therefore both a practical concern and a theoretically challenging problem. Given this vulnerability, a natural question arises: ***Does there exist any subspace, whether in weight space or activation space, that uniquely encodes safety alignment***? If safety is a distinct and structured property of the model, then updates or representations affecting it might consistently concentrate in identifiable geometric regions. This motivates a broader question: can we isolate or characterize safety-relevant subspaces that amplify aligned behavior or suppress harmful outputs?

To explore this question, we conduct four experiments probing the geometry of safety-related behavior across model weights and internal representations. We begin by analyzing FT updates from purely useful and harmful datasets, projecting them into subspaces derived from the alignment matrix to test whether safety correlates with energy or behavioral impact. Next, we examine contaminated FT, where a small fraction of harmful samples is mixed into a benign dataset. By projecting updates into the orthogonal complement of alignment-derived subspaces, we test whether harmful components can be selectively removed (see Figure 1). In the third experiment, we directly compare the dominant subspaces of useful, harmful, and alignment updates to assess whether safety-altering updates share consistent structure. Finally, we inspect the representation space, comparing internal activations from useful and harmful prompts to ask whether safety-related inputs occupy distinct subspaces, even when weight updates do not.

Across all experiments, we observe a consistent and surprising result: *no subspace, whether defined by alignment directions, update energies, or input representations, captures safety-specific behavior in isolation*. While certain subspaces, such as the top alignment directions, are behaviorally impactful, they amplify both helpful and harmful behaviors equally, reflecting general update sensitivity rather than alignment. Similarly, activations from harmful and helpful prompts occupy overlapping regions of representation space, offering no evidence for distinct "safety activation" geometry. These findings point to a fundamental limitation of subspace-based alignment strategies. If safe and unsafe behaviors cannot be cleanly separated geometrically, then projection- or filtering-based defenses are unlikely to suppress harmfulness without incurring equivalent losses in utility. Our key contributions are:

- We show that subspaces derived from alignment updates are not safety-specific; they amplify both helpful and harmful behaviors equally, reflecting general update sensitivity rather than alignment.
- We find that orthogonal projection intended to filter harmful updates leads to proportional losses in utility, suggesting no selective geometric separation between safe and unsafe behavior.
- We demonstrate that harmful and aligned updates do not share a consistent subspace, and that harmful prompts do not activate distinct regions of representation space.
- Through consistent results across five open-source LLMs evaluated in multiple experiments, we challenge the view that safety alignment is geometrically localized and reveal fundamental limitations of subspace-based defenses.

2 Preliminaries

Notation. Let $\mathbf{W}_0$ denote the parameters of the *base* model, and let $\mathbf{W}_\mathrm{A}$ represent the parameters of the *aligned* instruction-tuned model. We further fine-tune the aligned model on a task-specific dataset $\mathcal{D}j$, where $j \in \{\mathrm{Useful}, \mathrm{Harmful}, \mathrm{Contaminated}\}$, resulting in parameters $\mathbf{W}_{\mathrm{FT},j}$. We decompose the total parameter update as the sum of two components:

$$\Delta_\mathrm{A} := \mathbf{W}_\mathrm{A} - \mathbf{W}_0 \qquad \text{(alignment update)}, \tag{1}$$

$$\Delta_\mathrm{T}^j := \mathbf{W}_{\mathrm{FT},j} - \mathbf{W}_\mathrm{A} \qquad \text{(task-specific update)}. \tag{2}$$

Importance of Alignment Directions (Δ_A). Alignment training typically emphasizes behavioral properties such as harmlessness, helpfulness, and honesty. Empirical studies (10; 18; 37) suggest that the alignment update Δ_A encodes directions in parameter space that are strongly correlated with these safety attributes. Our goal is to systematically control the extent to which the subsequent task-specific update Δ_{T}^{j} interacts with these alignment directions.

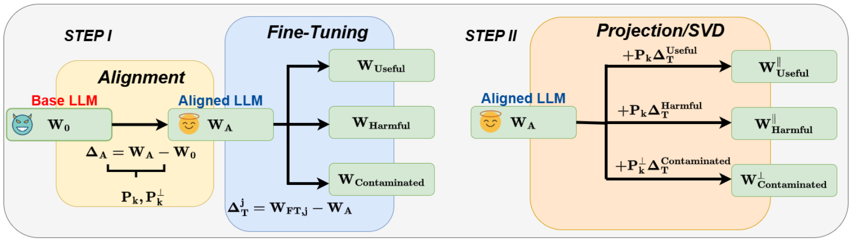

Figure 1: The base model W_o is instruction-tuned to produce the aligned model W_A. **Step 1:** The difference $\Delta_A = W_A - W_o$ defines a *safety direction*, from which projection matrices P_k (top-K subspace) and $P_k^{\perp}$ (orthogonal subspace) are derived. W_A is then fine-tuned on three datasets: helpful, harmful, and contaminated, to yield W_{useful}, W_{harmful}, and $W_{\text{contaminated}}$, with updates Δ_{t_j}. **Step 2:** Project Δ_{t_j} using P_k and $P_k^{\perp}$, and add back to W_A to obtain projected models for evaluation. In addition, SVD is performed on the task-specific updates and MSO is computed between the top-K singular vectors.

Constructing the Alignment Subspace. To formalize this notion, we begin by constructing the alignment subspace. Each tensor in the alignment update Δ_{A} is first reshaped into a matrix (flattened if needed) $V_{\mathrm{A}} \in \mathbb{R}^{M \times N}$. We perform a thin singular value decomposition (SVD) of the form $V_{\mathrm{A}} = U\Sigma V^{\top}$, which reveals the principal directions of parameter change (12; 36), ranked by their contribution to the Frobenius norm. The top k (**Top-K**) right singular vectors in V are then selected to define the *alignment subspace*:

$$\mathcal{S}_k := \operatorname{span}(U_k), \quad U_k \in \mathbb{R}^{M \times k}, \quad k \leq \operatorname{rank}(V_{\mathrm{A}}). \tag{3}$$

Intuitively, $\mathcal{S}_k$ captures the k most significant directions of parameter shifts resulting from alignment training. The alignment subspace naturally induces projection operators:

$$P_k := U_k U_k^{\top}, \quad P_k^{\perp} := I - P_k, \tag{4}$$

where P_k projects a matrix onto the alignment subspace, and $P_k^{\perp}$ onto its orthogonal complement.

Projection Schemes. Given a fractional rank hyperparameter $\varrho \in (0, 1]$, we determine $k = \lfloor \varrho \cdot \min(M, N) \rfloor$ and apply one of two projection-based update schemes to the task-specific update:

$$\textbf{Parallel}: \quad \tilde{\Delta}_{\mathrm{T}}^{j} = P_k \Delta_{\mathrm{T}}^{j}, \quad \mathbf{W}_{\text{parallel}} = \mathbf{W}_{\mathrm{A}} + \tilde{\Delta}_{\mathrm{T}}^{j}, \tag{5}$$

$$\textbf{Orthogonal}: \quad \tilde{\Delta}_{\mathrm{T}}^{j} = P_k^{\perp} \Delta_{\mathrm{T}}^{j}, \quad \mathbf{W}_{\text{orthogonal}} = \mathbf{W}_{\mathrm{A}} + \tilde{\Delta}_{\mathrm{T}}^{j}. \tag{6}$$

Equation 5 retains the update components that align with the alignment directions, while Equation 6 removes this aligned component, retaining only the update orthogonal to the alignment subspace.

Control Experiments. To further assess the specificity and effectiveness of the alignment subspace, we introduce two control experiments:

- **Random-K:** Instead of using the top-k singular vectors from the SVD of V_{A}, we randomly sample k singular vectors from the full set to construct a randomized alignment subspace.
- **Random:** We replace V_{A} with a random matrix of the same dimensions, perform its SVD, and use the top-k singular vectors to define a synthetic alignment subspace.

Energy-Kept Ratio. We introduce the fractional energy metric to quantify the extent of overlap between the task update and the alignment subspace:

$$\mathcal{E}_k(\Delta_{\mathrm{T}}^{j}) := \frac{\|P_k \Delta_{\mathrm{T}}^{j}\|_F^2}{\|\Delta_{\mathrm{T}}^{j}\|_F^2}, \quad \mathcal{E}_k^{\perp}(\Delta_{\mathrm{T}}^{j}) = 1 - \mathcal{E}_k(\Delta_{\mathrm{T}}^{j}). \tag{7}$$

Mode Subspace Overlap (MSO). Let $\mathbf{V} \in \mathbb{R}^{d \times n_V}$ and $\mathbf{W} \in \mathbb{R}^{d \times n_W}$ be two matrices with a shared ambient dimension d but possibly different column counts. We extract their principal directions using thin SVD:

$$\mathbf{V} = U_V \Sigma_V V_V^\top, \quad \mathbf{W} = U_W \Sigma_W V_W^\top. \tag{8}$$

For a chosen energy-retention fraction $\eta \in (0, 1]$, we select the smallest k_V and k_W such that the top k_V (resp. k_W) left singular vectors capture at least an η-fraction of $\|\Sigma_V\|_F^2$ (resp. $\|\Sigma_W\|_F^2$). This yields orthonormal bases $Q_V \in \mathbb{R}^{d \times k_V}$ and $Q_W \in \mathbb{R}^{d \times k_W}$. The *overlap matrix* is then defined as:

$$S = Q_V^\top Q_W \in \mathbb{R}^{k_V \times k_W}. \tag{9}$$

To quantify the similarity between these η-energy subspaces, we use MSO metric:

$$\mathrm{MSO}(\mathbf{V}, \mathbf{W}; \eta) = \frac{\|S\|_F^2}{\min(k_V, k_W)}, \quad 0 \leq \mathrm{MSO} \leq 1. \tag{10}$$

Intuitively, $\mathrm{MSO}(\mathbf{V}, \mathbf{W}; \eta)$ measures the overlap between the top-η energy components of $\mathbf{V}$ and $\mathbf{W}$: it equals 0 for orthogonal subspaces and 1 for identical spans. As a baseline, the expected overlap between random subspaces of dimensions k_V and k_W in $\mathbb{R}^d$ is given analytically by:

$$\mathbb{E}[\mathrm{overlap}] = \frac{\max(k_V, k_W)}{d}. \tag{11}$$

Models Used. Throughout our work, we evaluate both base and instruction-aligned versions of several open-source LLMs. For example, we consider Qwen-2.5 3B (base) alongside its aligned variant, Qwen-2.5 3B Instruct. We report results for base and aligned versions of five models: LLaMA 3.2 1B (11), LLaMA 2 7B (51), Qwen-2.5 1B (58), Qwen-2.5 3B, and Qwen-2.5 7B.

3 Do Alignment Subspaces Encode Safety?

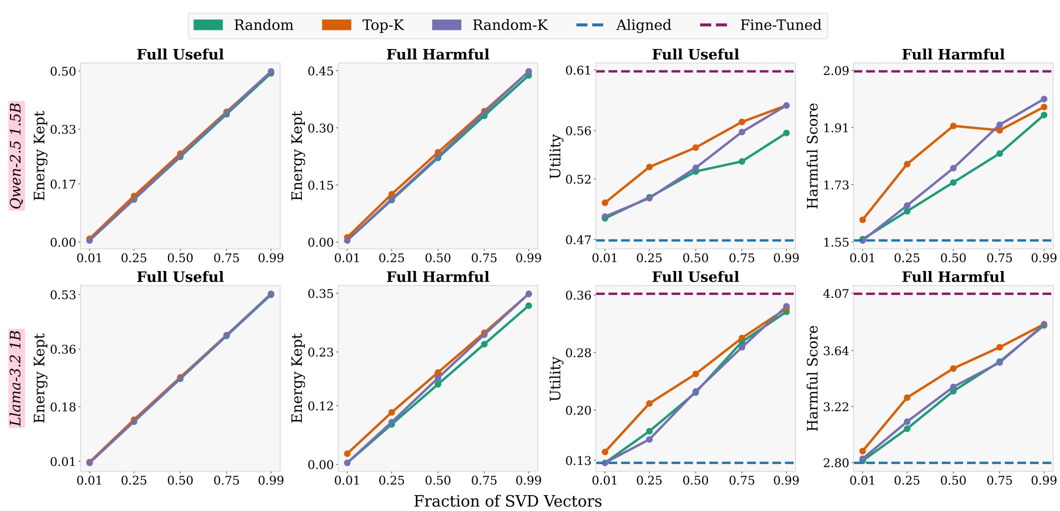

Figure 2: Parallel projection-based update schemes across varying SVD fractions. We report the energy-kept ratio for models fine-tuned on Full Useful and Full Harmful data, utility for models fine-tuned on Full Useful, and harmfulness for models fine-tuned on Full Harmful.

A central question in understanding safety alignment is whether specific directions in weight space, such as those defined by the difference between a base model and its RLHF-aligned counterpart, encode information unique to safety. If this is the case, then constraining FT updates to lie within these subspaces could offer a principled way to guard against harmful optimization. We begin our investigation by examining whether task-specific FT updates align differently with the top directions of the alignment matrix, depending on whether the task is helpful or harmful.

Experimental Setup. We fine-tune an aligned instruction-tuned model on two distinct datasets. The first is a 20K subset of MetaMathQA (63), a benchmark of math word problems representing a useful task without safety concerns. The second is a 4K unsafe subset of BeaverTails (26), a synthetic dataset of harmful instruction–response pairs designed to elicit unsafe behavior. We denote the resulting weight updates as Δ_T^{Useful} and $\Delta_T^{\text{Harmful}}$ respectively. To quantify behavioral effects, we evaluate harmfulness using the AdvBench dataset (68), with GPT-4o-mini (24) scoring each response from 1 (least harmful) to 5 (most harmful); the final score is the average across samples. Utility is measured by accuracy on the GSM8k test set (8), using final answer correctness. We compute these metrics, energy-kept ratio, utility, and harmfulness, for the projected models $\mathbf{W}_{\text{parallel}}$ and $\mathbf{W}_{\text{orthogonal}}$, as well as for the base, aligned, fine-tuned, and control models.

Results: Energy Is Uniform Across Subspaces, Performance Is Not. As shown in Figure 2, the fraction of energy retained in projected updates increases linearly with subspace rank and is consistent across all three subspace types. This pattern holds for both helpful and harmful updates. There is no evidence that update energy is preferentially concentrated in the top directions of Δ_A for safe vs unsafe FT. This suggests that if a "safety subspace" exists, it is not captured simply by energetic alignment with the dominant directions of Δ_A. However, while energy is evenly distributed, behavioral impact is not. We can observe that projecting Δ_T^{Useful} onto the top-k directions consistently improves utility relative to random projections with equal energy, in Figure 2 and Table 1. Similarly, projecting $\Delta_T^{\text{Harmful}}$ onto the same directions increases harmfulness. Thus, the top singular directions of Δ_A are not uniquely aligned with safety, but they are generally potent. Updates along these directions are more effective, whether the goal is to enhance utility or to elicit harmful behavior. We present results on all models in Table 3 (Appendix B).

Table 1: Parallel projection-based update schemes across varying SVD fractions. We report the utility for models fine-tuned on Full Useful data, and harmfulness for models fine-tuned on Full Harmful.

Model	**Method**	**Utility (↑)**							**Harmful Score (↓)**						
		Aligned	**SVD Fractions**					**FT**	**Aligned**	**SVD Fractions**					**FT**
			0.01	**0.25**	**0.50**	**0.75**	**0.99**			**0.01**	**0.25**	**0.50**	**0.75**	**0.99**	
Qwen-2.5 1.5B	Top-K	0.47	0.50	0.53	0.55	0.57	0.58	0.61	1.55	1.62	1.80	1.92	1.90	1.97	2.09
	Random-K	0.47	0.49	0.50	0.53	0.56	0.58	0.61	1.55	1.55	1.66	1.78	1.92	2.00	2.09
	Random	0.47	0.49	0.50	0.53	0.53	0.56	0.61	1.55	1.56	1.65	1.74	1.83	1.95	2.09
Llama-3.2 1B	Top-K	0.13	0.14	0.21	0.25	0.30	0.34	0.36	2.80	2.89	3.29	3.51	3.66	3.84	4.07
	Random-K	0.13	0.13	0.16	0.23	0.29	0.34	0.36	2.80	2.83	3.11	3.37	3.55	3.84	4.07
	Random	0.13	0.13	0.17	0.22	0.29	0.34	0.36	2.80	2.81	3.05	3.34	3.56	3.83	4.07

Implications: Alignment Directions Reflect Update Sensitivity, Not Safety. This symmetry across tasks is important. The fact that top-k directions amplify both helpful and harmful behavior equally suggests they do not encode alignment directly. Instead, they appear to represent axes of general parameter sensitivity, directions where updates tend to induce large changes in model behavior. In this sense, Δ_A captures a general learning geometry: directions that are especially effective for optimization, not inherently safe. We draw three key takeaways. First, neither helpful nor harmful updates preferentially align with the top subspaces of Δ_A in terms of energy. Second, those same subspaces are more behaviorally expressive, enhancing both utility and harmfulness depending on the task. Third, this challenges the notion that Δ_A encodes safety-specific information. Its dominant directions support effective learning broadly, without guiding its ethical character. Thus, using Δ_A to constrain updates may regulate the magnitude of behavior change, but not its direction or valence.

4 Can Harmful Subspaces Be Removed?

Having analyzed helpful and harmful updates in isolation, we now consider a more realistic scenario: contaminated FT. This involves adding a small fraction of harmful examples to an otherwise benign dataset, producing updates that blend both signals. Contaminated data is particularly dangerous because it can degrade alignment without obvious signs. Prior work shows that even limited contamination can erode safety, causing models to revert to unsafe behaviors. While earlier experiments identified expressive subspaces, we now ask the reverse: can we remove harmful components from an update? We test whether filtering specific subspaces, particularly those aligned with the dominant directions of the alignment matrix, can reduce harmfulness while preserving utility.

Experimental Setup. We construct a contaminated dataset by mixing 20% harmful data from BeaverTails with 80% of the 20K MetaMathQA subset. FT on this mixture yields a single contaminated update, Δ_T. To suppress harmful behavior, we apply the orthogonal projection strategy from Section 2, removing components along the top-k alignment directions. Specifically, we compute $\tilde{\Delta}_T = P_k^{\perp} \Delta_T$, where $P_k^{\perp}$ projects onto the complement of the alignment subspace. We evaluate the resulting models on GSM8K (utility) and AdvBench (harmfulness). Our goal is to test whether removing alignment-aligned components can reduce harmfulness while preserving task performance.

Results: Utility And Harmfulness Drop Together. Figure 3 shows the effects of orthogonal projection on retained energy, utility, and harmfulness. As k increases, implying more of the update is removed, the retained energy declines steadily across all projection types (random, top-k, and random-k). Utility and harmfulness scores (Figure 3, Table 2) follow a similar downward trend. However, the rate of decline differs by projection strategy. Removing top-k alignment components reduces utility more sharply than random projections. At the same time, harmfulness decreases at a similar rate, indicating no selective suppression of harmful behavior. In effect, safety improvements come at a proportional cost to task performance, with no clear advantage in targeting the alignment subspace. We present results on all models in Table 4 (Appendix C).

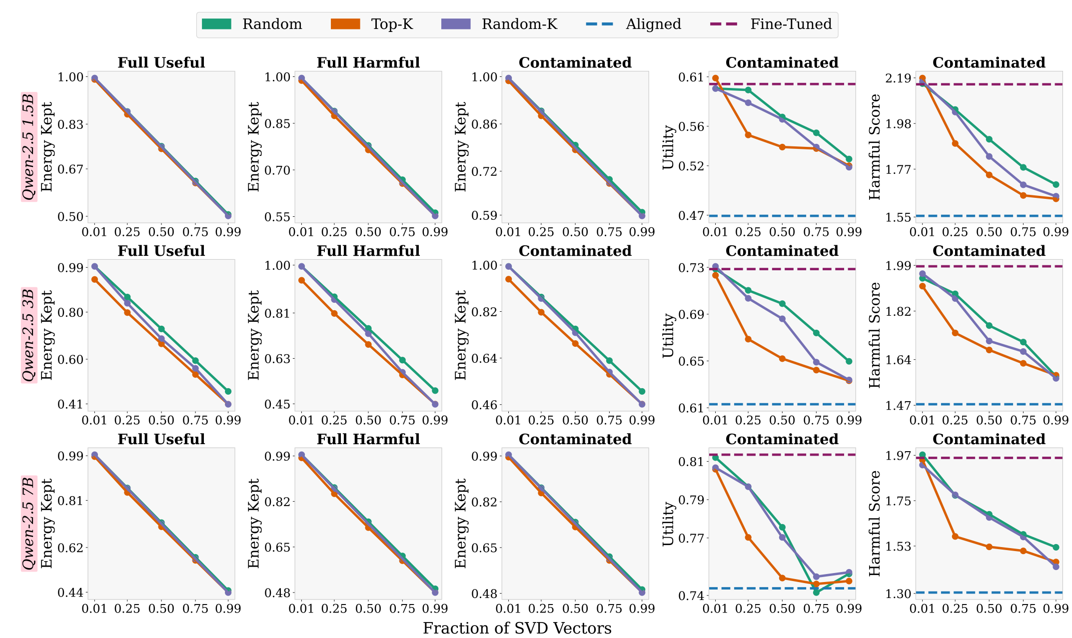

Figure 3: Parallel projection-based update schemes across varying SVD fractions. We report the energy-kept ratio for models fine-tuned on Full Useful, Full Harmful and Contaminated data; and utility and harmfulness for models fine-tuned on Contaminated.

Implications: No Selective Removal Is Possible. These results indicate that the alignment subspace does not uniquely encode safety or harmfulness but rather captures directions broadly important for learning. Removing these directions degrades both utility and harmfulness at similar rates. If harmful behavior were confined to distinct subspaces, we would expect a steeper drop in harmfulness than utility, yet this is not observed. Even if safety-relevant directions exist, they are not recoverable from the alignment matrix alone, especially under contamination. The update blends helpful and harmful objectives, making

Table 2: Parallel projection-based update schemes across varying SVD fractions. We report the utility and harmfulness for models fine-tuned on Contaminated data.

Model	Method	Utility (↑) Aligned	Utility SVD Fractions 0.01	0.25	0.50	0.75	0.99	Utility FT	Harmful Score (↓) Aligned	Harmful SVD Fractions 0.01	0.25	0.50	0.75	0.99	Harmful FT
Qwen-2.5 1.5B	Top-K	0.47	0.50	0.53	0.55	0.57	0.58	0.60	1.55	1.58	1.65	1.80	1.91	1.92	2.16
	Random-K	0.47	0.49	0.52	0.53	0.55	0.55	0.60	1.55	1.56	1.62	1.63	1.87	1.92	2.16
	Random	0.47	0.49	0.50	0.52	0.52	0.54	0.61	1.55	1.58	1.64	1.68	1.74	1.92	2.16
Qwen-2.5 3B	Top-K	0.61	0.63	0.64	0.65	0.68	0.69	0.73	1.47	1.49	1.58	1.69	1.76	1.83	1.99
	Random-K	0.61	0.62	0.64	0.64	0.66	0.69	0.73	1.47	1.45	1.55	1.62	1.65	1.91	1.99
	Random	0.61	0.62	0.63	0.64	0.65	0.68	0.73	1.47	1.45	1.50	1.57	1.75	1.83	1.99
Qwen-2.5 7B	Top-K	0.74	0.74	0.75	0.75	0.75	0.78	0.81	1.30	1.31	1.56	1.60	1.68	1.67	1.96
	Random-K	0.74	0.74	0.75	0.76	0.75	0.78	0.81	1.30	1.35	1.41	1.46	1.59	1.67	1.96
	Random	0.74	0.74	0.75	0.75	0.75	0.78	0.81	1.30	1.34	1.40	1.48	1.56	1.63	1.96

its projection agnostic to intent. As a result, orthogonal projection fails to selectively suppress harmful behavior. Subspace filtering based on alignment directions imposes a strict tradeoff: gains in safety come with proportional utility loss. This challenges the effectiveness of subspace-based defenses under contaminated FT.

5 Are Safety Weight Subspaces Distinct?

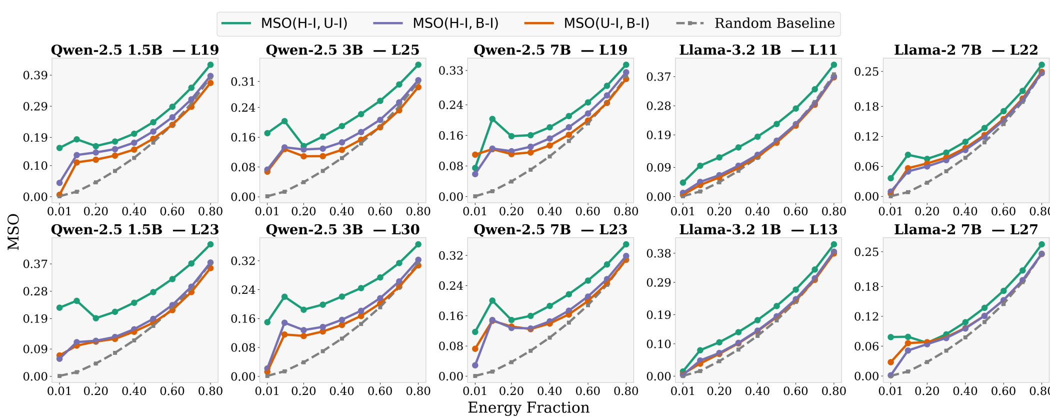

Figure 4: Mode Subspace Overlap (MSO) at the 70- and 85- percentile layers for pairwise comparisons of the dominant subspaces from Harmful fine-tuned (H), Instruction-tuned (I), and Base (B) models.

A natural question is whether a dedicated region of parameter space, what we might call a *safety subspace*, captures safety-specific behavior. Such a subspace should meet two criteria: (i) safety-relevant updates, whether from alignment or harmful FT, should lie significantly within it; and (ii) task-specific updates unrelated to safety should have minimal overlap, with projections onto the subspace leaving model safety unchanged. Crucially, these properties must generalize across tasks and datasets to rule out dataset-specific artifacts. Our earlier results argue against the top subspaces of the alignment matrix meeting these criteria. These directions are highly sensitive to any update, helpful or harmful, but do not isolate safety. Still, it remains open whether some other set of directions, possibly outside the alignment subspace, could fulfill this role. To explore this, we directly compare the dominant subspaces of different update types.

Experimental Setup. We compare the principal subspaces of 3 updates: the alignment update Δ_A (from base to aligned model), the harmful FT update $\Delta_T^{\text{Harmful}}$ (trained on BeaverTails), and the useful update Δ_T^{Useful} (trained on a 20K subset of MetaMathQA). Notably, the negated alignment update, $-\Delta_A$, reverses alignment by pushing the model back toward its unaligned base state, effectively acting as a harmful update and serving as a useful reference. For a given energy threshold $\eta \in (0, 1]$, we compute $\mathrm{MSO}(\cdot, \cdot; \eta)$ (Section 2 for three pairs: (i) $\left(\Delta_{\mathrm{T}}^{\mathrm{Useful}}, \Delta_{\mathrm{T}}^{\mathrm{Harmful}}\right)$, to assess whether helpful and harmful FT affect similar subspaces; (ii) $\left(\Delta_{\mathrm{T}}^{\mathrm{Useful}}, -\Delta_A\right)$, to test alignment between helpful updates and reversed alignment; and (iii) $\left(\Delta_{\mathrm{T}}^{\mathrm{Harmful}}, -\Delta_A\right)$, to compare two harmful directions. We sweep over η, with small values isolating high-energy directions and larger values approaching full-rank overlap. We include the random-subspace baseline $\max(k_V, k_W)/d$; values above this baseline indicate significant geometric alignment, while values near it suggest chance-level overlap.

Results: Representations Overlap Across Tasks. Figure 4 shows the pairwise overlap between the dominant subspaces (top-k directions) of each update. All pairs exhibit greater overlap than random baselines, indicating shared structure. However, the strongest overlap is between the useful and harmful updates, not between alignment and harmful updates, as one might expect if safety were a shared component. This is a key finding. If a safety subspace existed, it would likely appear in the shared directions between alignment and harmful updates, which affect safety in opposite ways. This lack of substantial overlap suggests that no consistent, linear safety-specific subspace exists.

Implications: Shared Subspaces Drive Behavior, Not Safety. Together with earlier results, these findings suggest that safety-relevant updates do not lie in a well-defined or isolatable subspace. Instead, both alignment and harmfulness operate over complex, task-dependent, and likely non-linear directions. The high overlap between harmful and helpful update subspaces supports our earlier hypothesis: these directions form a general *learning subspace*, expressive across tasks but agnostic to safety. We find no evidence for a distinct safety subspace. Updates that influence safety, positively or negatively, do not share dominant directions. Any shared structure reflects general learning capacity rather than safety-specific behavior. As such, geometric separation of alignment remains elusive, and linear subspace methods cannot cleanly isolate safety in parameter space.

6 Do Safety Subspaces Exist In Representation Space?

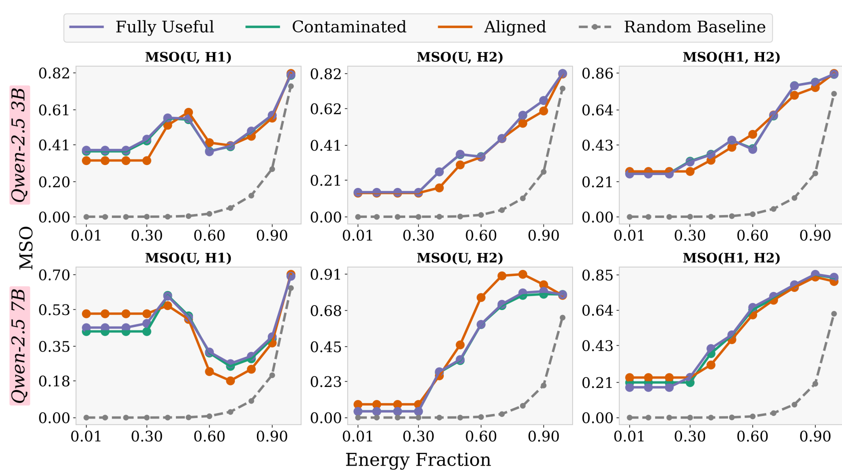

Figure 5: Average Mode Subspace Overlap (MSO) across layers in the 65–90% depth range for pairwise comparisons of activations from Useful (U) and multiple Harmful (H1, H2) prompt sets.

So far, our analysis has focused on the weight space, probing whether certain update directions correspond to safety-related behavior. Across experiments, we found no evidence for distinct subspaces encoding safety at the parameter level. However, safety may instead manifest through how inputs interact with the model, specifically, through the regions of representation space they activate. This motivates a final question: do safety-relevant inputs elicit distinct activation patterns, even if their corresponding weight updates overlap? While weight updates may distribute energy broadly, inputs could selectively activate specific directions. This perspective also offers a possible explanation for earlier results: even low-energy projections onto alignment directions produced strong behavioral effects, likely because inputs activated those directions disproportionately.

Experimental Setup. We compare internal activations induced by different prompt categories. Specifically, we pass useful (benign) prompts from the MATH dataset (17) and harmful prompts from BeaverTails (test set) and ToxiGen (14) through three models: the aligned model, the useful fine-tuned model, and the contaminated fine-tuned model. For each prompt, we record the hidden state of the *last* generated token at each transformer layer $\ell \in 0, \ldots, L$. At each layer, these hidden states are stacked into activation matrices of shape $\mathbb{R}^{n \times d}$, where d is the model's hidden size and n is the number of prompts (5000 for each dataset). We compute MSO (see Section 2) between activation matrices corresponding to the prompts from different datasets, sweeping over energy thresholds η. Lower values of η capture high-energy activation modes, while higher values approximate full-rank comparisons. We plot MSO curves alongside the random-subspace baseline $\max(k_{\text{Useful}}, k_{\text{Harmful}})/d$, and report averages over layers in the 65–90 % depth percentile.

Results: Representation Subspaces Overlap Across Tasks. Figure 5 reports MSO values across all pairs of prompt categories. Useful and harmful prompts consistently exhibit overlap above the random baseline, indicating activation of shared high-energy subspaces in representation space. Notably, the overlap between the two harmful prompt sets is not consistently higher than their overlap with helpful prompts; in some cases, the useful–harmful overlap is greater than the harmful–harmful one. The degree of overlap also varies across model configurations. Some models show strong alignment even in the top subspaces, while others exhibit more gradual increases in overlap, becoming significant only at higher energy thresholds. This variability suggests that representational similarity is influenced more by model-specific factors than by the safety content of the prompts alone. Results on more models are provided in Figure 6 (Appendix D).

Implications: Shared Subspaces Drive Behavior, Not Safety. These observations suggest that while all prompt types activate shared subspaces more than expected by chance, there is no evidence of a distinct safety-violating subspace. If such a subspace existed, activations from harmful prompts would consistently exhibit greater mutual overlap than with useful prompts, which is not the case. Instead, the results indicate that prompts with differing safety implications are processed through broadly overlapping representations. This supports our earlier hypothesis: the directions most responsible for driving behavior reflect general-purpose representational subspaces rather than safety-specific ones. These directions are activated across tasks and prompt types, implying that LLMs do not internally separate "safe" and "unsafe" activation modes, but instead rely on shared, high-impact subspaces. We find no evidence of a distinct safety subspace in representation space. Useful and harmful prompts show substantial overlap, even across prompt sets with very different behavioral consequences. Combined with our findings in weight space, these results suggest that both aligned and harmful behaviors emerge from shared representational mechanisms rather than separable subspaces.

7 Conclusion

This work set out to investigate how safety alignment is encoded in LLMs, and whether it can be isolated through geometric structure in weight or activation space. Motivated by the challenge of preserving alignment under continued fine-tuning, particularly in adversarial or contaminated settings, we conducted a systematic study across four experiments and five open-source LLMs, examining both parameter updates and internal representations. Our findings challenge the common assumption that alignment corresponds to safety-specific subspaces. Subspaces with high behavioral impact are not unique to alignment; they enhance both utility and harmfulness, and their removal degrades both. This indicates that these directions reflect general-purpose learning rather than safety alone. Moreover, harmful and helpful prompts activate overlapping regions of representation space, offering no evidence for distinct "safety activation" geometry. Together, these results suggest that safety alignment is not cleanly separable in geometric terms. While this complicates subspace-based defenses, it also highlights the potential of high-impact directions, if appropriately constrained, for guiding both safe fine-tuning and activation-level control. More broadly, our work calls for rethinking geometric assumptions in interpretability and alignment, and for developing methods that engage with the entangled nature of learned representations.

Acknowledgements and Disclosure of Funding

This work received support from Mohamed bin Zayed University of Artificial Intelligence (MBZUAI) and was partially funded by the ADIA Lab Fellowship.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] 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, 2024.

[3] Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic, 2024.

[4] Rishabh Bhardwaj and Soujanya Poria. Language model unalignment: Parametric red-teaming to expose hidden harms and biases, 2023.

[5] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions, 2024.

[6] Zehua Cheng, Manying Zhang, Jiahao Sun, and Wei Dai. On weaponization-resistant large language models with prospect theoretic alignment. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10309–10324, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[7] Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. Safety-aware fine-tuning of large language models. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[9] Xander Davies, Eric Winsor, Tomek Korbak, Alexandra Souly, Robert Kirk, Christian Schroeder de Witt, and Yarin Gal. Fundamental limitations in defending llm finetuning apis, 2025.

[10] Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. Safemerge: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging, 2025.

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[12] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[13] Francisco Eiras, Aleksandar Petrov, Philip Torr, M. Pawan Kumar, and Adel Bibi. Do as i do (safely): Mitigating task-specific fine-tuning risks in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

[15] Will Hawkins, Brent Mittelstadt, and Chris Russell. The effect of fine-tuning on language model toxicity, 2024.

[16] Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety, 2024.

[17] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[18] 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 fine-tuning large language models, 2025.

[19] Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning, 2024.

[20] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey, 2024.

[21] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack, 2024.

[22] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation, 2025.

[23] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Virus: Harmful fine-tuning attack for large language models bypassing guardrail moderation, 2025.

[24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[25] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023.

[26] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[27] Joshua Kazdan, Lisa Yu, Rylan Schaeffer, Chris Cundy, Sanmi Koyejo, and Krishnamurthy Dvijotham. No, of course i can! refusal mechanisms can be exploited using harmless fine-tuning data, 2025.

[28] Connor Kissane, robertzk, Arthur Conmy, and Neel Nanda. Open source replication of anthropic's crosscoder paper for model-diffing. AI Alignment Forum post, Oct 2024. Accessed 2025-05-15.

[29] Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b, 2024.

[30] Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. Salora: Safety-alignment preserved low-rank adaptation, 2025.

[31] Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to LLM security. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Guozhi Liu, Weiwei Lin, Tiansheng Huang, Ruichao Mo, Qi Mu, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation, 2025.

[33] Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. Robustifying safety-aligned large language models through clean data curation, 2024.

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[35] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates, 2025.

[36] Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The quarterly journal of mathematics*, 11(1):50–59, 1960.

[37] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[39] ShengYun Peng, Pin-Yu Chen, Matthew Hull, and Duen Horng Chau. Navigating the safety landscape: Measuring risks in finetuning large language models, 2024.

[40] Samuele Poppi, Zheng-Xin Yong, Yifei He, Bobbie Chern, Han Zhao, Aobo Yang, and Jianfeng Chi. Towards understanding the fragility of multilingual llms against fine-tuning attacks, 2025.

[41] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep, 2024.

[42] 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!, 2023.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024.

[45] Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning, 2024.

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[47] Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations*, 2025.

[48] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight llms, 2025.

[49] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[50] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[51] 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.

[52] Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment, 2024.

[53] Yibo Wang, Tiansheng Huang, Li Shen, Huanjin Yao, Haotian Luo, Rui Liu, Naiqiang Tan, Jiaxing Huang, and Dacheng Tao. Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation, 2025.

[54] 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, 2024.

[55] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022.

[56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[57] Di Wu, Xin Lu, Yanyan Zhao, and Bing Qin. Separate the wheat from the chaff: A post-hoc approach to safety re-alignment for fine-tuned language models, 2025.

[58] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[59] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2023.

[60] Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9236–9260, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[61] Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning, 2024.

[62] Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. A safety re-alignment framework via subspace-oriented model fusion for large language models, 2024.

[63] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024.

[64] Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing RLHF protections in GPT-4 via fine-tuning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 681–687, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[65] Wenxuan Zhang, Philip Torr, Mohamed Elhoseiny, and Adel Bibi. Bi-factorial preference optimization: Balancing safety-helpfulness in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[66] Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*, 2025.

[67] Minjun Zhu, Linyi Yang, Yifan Wei, Ningyu Zhang, and Yue Zhang. Locking down the finetuned llms safety, 2024.

[68] 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 Related Work

Safety Alignment and Task-Specific Fine-Tuning in LLMs. Large Language Models (LLMs) do not inherently follow instructions and often exhibit socially undesirable behaviors. To address this, various post-training methods, instruction-tuning and reinforcement learning from human feedback, are applied to align base LLMs with human values and improve their instruction-following capabilities (37; 44; 46; 55). However, studies have shown that fine-tuning these aligned models on harmful data can undo this alignment, restoring their original, socially unacceptable behaviors (59). This unalignment phenomenon has been demonstrated in both open-source models (29; 60) and proprietary models (4; 42; 64) via publicly available fine-tuning APIs, thereby exposing a new attack surface (9; 23; 27). Moreover, even fine-tuning on benign downstream tasks can degrade alignment (15; 16).

Defense Methods. To safeguard aligned LLMs against unalignment during fine-tuning, defenses have been proposed at three stages of the pipeline: the alignment stage, the fine-tuning stage, and the post-processing stage. The effectiveness of these defense methods is evaluated using downstream model utility and harmfulness (20).

Alignment Stage Defenses. Alignment stage defenses update the initial instruction-tuning process to ensure that downstream fine-tuning cannot easily overwrite the model's safety behavior. One approach augments the alignment loss, making harmful representations harder to recover during fine-tuning updates (45). Another line of work relies on safety-oriented data curation to preserve alignment under downstream fine-tuning(33). Adversarial and meta-learning techniques have also been combined to develop tamper-resistant methods that prevent harmfulness while maintaining task performance (48). A separate strategy introduces a regularization term to the alignment loss, which has been shown to preserve safety after fine-tuning (22). Perturbing safety-critical layers during instruction-tuning has also been shown to protect alignment (32). Additional work traces unalignment to excessive dependence on maximum-likelihood training, motivating an integrity preserving variant of this method (6). A study on "shallow alignment" also shows that instruction-tuning influences only the first few output tokens, whereas deeper alignment improves robustness (41).

Fine-Tuning Stage Defenses. Fine-tuning stage defenses modify the fine-tuning process to ensure that the model's alignment is preserved after update. One class of defenses focuses on data curation, augmenting the fine-tuning dataset to maintain alignment after update (5; 13). Another approach uses safety examples prefixed with a secret prompt, which act as backdoor triggers to reactivate safe behavior after fine-tuning (52). A data ranking based strategy has also been proposed, where low-quality data is down-ranked and high-quality data is up-ranked to better preserve safety (47). It has also been shown that prompt templates play an important role; removing the safety prompt during fine-tuning and reintroducing it at inference time can maintain alignment (35).

Optimization based defenses are another type of fine-tuning stage defenses. One line of work splits fine-tuning into an alignment phase and a utility phase, safeguarding both safety and task performance (21). Another approach combines safety and helpfulness objectives into a single loss (65).

Parameter level methods can also be used to preserve safety. One strategy identifies safety neurons and updates only those parameters (66). Another approach involves localizing safety layers and freezing their gradients, which has been shown to prevent unalignment (31). Another line of work explores constraining parameter changes to directions orthogonal to existing safety features, showing that this method preserves alignment (30). It has also been shown that harmful data can be filtered by matching fine-tuning embeddings against the top-k singular vectors of an activation matrix generated using a harmful dataset (7).

Post-Processing Stage Defenses. Post-processing stage defenses adjust the fine-tuned model to restore alignment and preserve usefulness. One approach adds a safety vector, defined as the difference between aligned and unaligned weights, to the fine-tuned parameters to regain safe behavior (3). Another line of work projects the fine-tuning update onto the alignment vector when their similarity drops below a threshold, or selectively merges layers from the fine-tuned and aligned models under the same criterion to achieve a similar effect (10; 18). A third strategy removes parameters identified as harmful after fine-tuning to restore alignment (19). It has been shown that safety directions in attention-head activations can also be located and used for targeted intervention (67) to realign the fine-tuned model. Another method detects update parameters whose signs contradict the original alignment and removes them (57). Additional work restores safety-critical neurons (61), fuses aligned and fine-tuned models (62), or adds an optimized post-hoc perturbation to recover alignment (53).

Safety Mechanisms in Fine-Tuned and Aligned LLMs. Recent studies have examined how LLMs express safety over neurons, layers, and activations. One study finds that safety related information is language agnostic, identifies parameters whose modification affects alignment, and shows that freezing these parameters during fine-tuning does not ensure safety (40). Another line of work locates sparse regions in parameter space whose removal weakens alignment, and likewise observes that freezing these regions alone is insufficient to maintain model alignment (54). A separate analysis maps a safety basin in weight space, noting that random perturbations inside the basin leave safety intact, whereas fine-tuning moves weights outside it (39). Finally, work on the activation residual stream isolates a refusal direction, removing this direction prevents refusal to harmful prompts, while adding it triggers refusal to benign ones (2).

B Do Alignment Subspaces Encode Safety?

We provide additional results in Table 3 to support the analysis presented in Section 3.

C Can Harmful Subspaces Be Removed?

Table 4 presents supplementary results that further substantiate the findings discussed in Section 4.

D Do Safety Subspaces Exist in Representation Space?

To complement the discussion in Section 5, we include extended results in Figure 6.

E Experimental Details

We implemented all experiments using PyTorch (38) and the HuggingFace Transformers library (56). We ran all experiments on a single NVIDIA A6000 GPU (48 GB). To save memory, all base models are initalized in `torch.bfloat16` precision. All models are trained using the AdamW optimizer (34). Detailed hyperparameter configurations for full fine-tuning of each model are presented in Table 5.

F Dataset Details

We use the **MetaMathQA** dataset (63) for fine-tuning, which reformulates existing math problems from alternative perspectives without introducing new content. To evaluate performance, we rely on the **GSM8K** benchmark (8), a dataset of elementary-level math questions that require multi-step reasoning. Models are assessed based solely on the correctness of the final numerical answer. For our activation-based analysis, we sample prompts from the **MATH** dataset (17), which contains challenging, competition-style arithmetic problems.

BeaverTails (26) is a valuable dataset for studying safety by independently annotating question–answer pairs for both helpfulness and harmlessness. We use the training set to

Table 3: Parallel projection-based update schemes across varying SVD fractions. We report the utility for models fine-tuned on Full Useful data, and harmfulness for models fine-tuned on Full Harmful.

Model	Method	Utility (↑) SVD Fractions					Harmful Score (↓) SVD Fractions				
		0.01	0.25	0.50	0.75	0.99	0.01	0.25	0.50	0.75	0.99
Qwen-2.5 1.5B	Base	0.21					3.27				
	Aligned	0.47					1.55				
	Fine-Tuned	0.61					2.09				
	Top-K	0.50	0.53	0.55	0.57	0.58	1.62	1.80	1.92	1.90	1.97
	Random-K	0.49	0.50	0.53	0.56	0.58	1.55	1.66	1.78	1.92	2.00
	Random	0.49	0.50	0.53	0.53	0.56	1.56	1.65	1.74	1.83	1.95
Llama-3.2 1B	Base	0.03					4.13				
	Aligned	0.13					2.80				
	Fine-Tuned	0.36					4.07				
	Top-K	0.14	0.21	0.25	0.30	0.34	2.89	3.29	3.51	3.66	3.84
	Random-K	0.13	0.16	0.23	0.29	0.34	2.83	3.11	3.37	3.55	3.84
	Random	0.13	0.17	0.22	0.29	0.34	2.81	3.05	3.34	3.56	3.83
Qwen-2.5 3B	Base	0.44					2.53				
	Aligned	0.61					1.47				
	Fine-Tuned	0.72					2.16				
	Top-K	0.63	0.64	0.65	0.68	0.69	1.48	1.71	1.81	1.91	1.92
	Random-K	0.62	0.63	0.64	0.65	0.69	1.44	1.55	1.62	1.74	1.91
	Random	0.62	0.63	0.64	0.65	0.68	1.44	1.50	1.66	1.75	1.83
Qwen-2.5 7B	Base	0.69					1.90				
	Aligned	0.74					1.30				
	Fine-Tuned	0.81					2.12				
	Top-K	0.72	0.74	0.76	0.77	0.77	1.34	1.56	1.66	1.76	1.84
	Random-K	0.73	0.75	0.74	0.75	0.77	1.34	1.44	1.53	1.64	1.84
	Random	0.74	0.75	0.75	0.76	0.76	1.33	1.40	1.48	1.56	1.75
Llama-2 7B	Base	0.05					4.27				
	Aligned	0.20					1.74				
	Fine-Tuned	0.30					3.41				
	Top-K	0.21	0.24	0.26	0.28	0.29	1.81	2.34	2.61	2.90	3.15
	Random-K	0.20	0.23	0.25	0.28	0.29	1.74	1.91	2.09	2.63	3.13
	Random	0.20	0.23	0.25	0.28	0.28	1.77	1.91	2.15	2.57	3.03

fine-tune models in both harmful and contaminated settings, and draw prompts from the test split for our activation-based experiments.

AdvBench (68) consists of 500 prompts designed to elicit a wide range of harmful behaviors, including profanity, threats, misinformation, discrimination, cybercrime, and other forms of dangerous or illegal content framed as instructions. We use this benchmark to quantify model harmfulness: higher success in responding to these prompts indicates greater unsafe behavior.

ToxiGen (14) is a large-scale dataset composed of both toxic and non-toxic statements. We use a subset of its prompts to analyze model activations in response to harmful content.

Table 4: Parallel projection-based update schemes across varying SVD fractions. We report the utility and harmfulness for models fine-tuned on Contaminated data.

Model	**Method**	**Utility (↑)** emphSVD Fractions 0.01	0.25	0.50	0.75	0.99	**Harmful Score (↓)** emphSVD Fractions 0.01	0.25	0.50	0.75	0.99
Qwen-2.5 1.5B	Base			0.21					3.27		
	Aligned			0.47					1.55		
	FT			0.60					2.16		
	Top-K	0.50	0.53	0.52	0.55	0.56	1.59	1.65	1.79	1.91	1.92
	Random-K	0.49	0.52	0.53	0.55	0.55	1.56	1.62	1.63	1.87	1.92
	Random	0.49	0.50	0.52	0.52	0.54	1.58	1.64	1.68	1.74	1.92
Llama-3.2 1B	Base			0.026					4.13		
	Aligned			0.13					2.80		
	FT			0.37					3.60		
	Top-K	0.14	0.20	0.25	0.29	0.33	2.84	2.90	3.05	3.36	3.45
	Random-K	0.13	0.16	0.22	0.29	0.33	2.81	2.90	3.03	3.19	3.45
	Random	0.13	0.16	0.22	0.28	0.33	2.84	2.90	3.19	3.19	3.45
Qwen-2.5 3B	Base			0.44					2.53		
	Aligned			0.61					1.47		
	FT			0.73					1.99		
	Top-K	0.62	0.63	0.65	0.68	0.69	1.49	1.58	1.69	1.76	1.83
	Random-K	0.62	0.64	0.64	0.66	0.69	1.45	1.55	1.62	1.65	1.91
	Random	0.62	0.63	0.64	0.65	0.68	1.45	1.50	1.57	1.75	1.83
Qwen-2.5 7B	Base			0.69					1.90		
	Aligned			0.74					1.30		
	FT			0.81					1.96		
	Top-K	0.74	0.75	0.75	0.75	0.78	1.30	1.55	1.60	1.68	1.67
	Random-K	0.74	0.75	0.76	0.75	0.78	1.35	1.41	1.46	1.59	1.67
	Random	0.74	0.75	0.75	0.75	0.78	1.34	1.40	1.48	1.56	1.63
Llama-2 7B	Base			0.053					4.27		
	Aligned			0.20					1.74		
	FT			0.30					3.08		
	Top-K	0.21	0.23	0.25	0.27	0.28	1.77	1.91	2.15	2.38	2.74
	Random-K	0.20	0.23	0.26	0.28	0.28	1.74	1.91	2.09	2.38	2.79
	Random	0.20	0.23	0.25	0.27	0.28	1.77	1.91	2.15	2.38	2.74

Table 5: Hyperparameter settings for fine-tuning the various models.

Optimizer	AdamW
Batch size	1
Max. Seq. Len	512
Grad Acc. Steps	32
Epochs	1
Learning Rate	1×10^{-5}
LR Scheduler	Cosine
Warmup Ratio	0.02

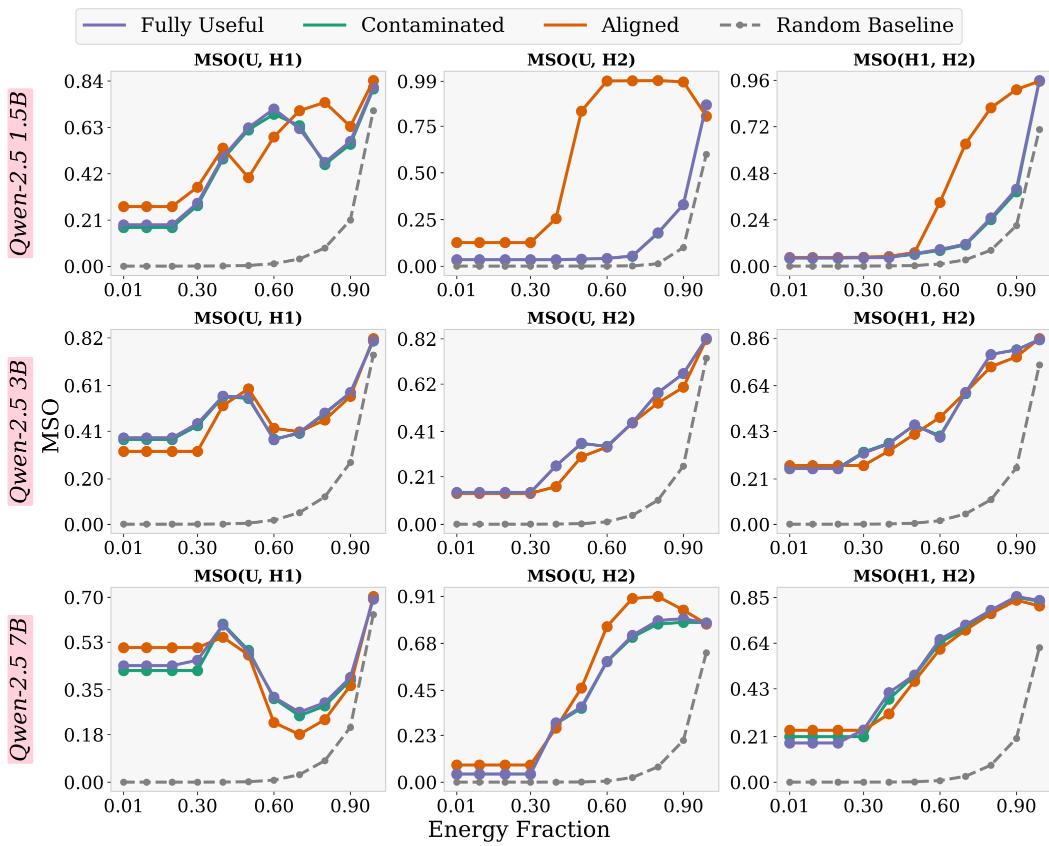

Figure 6: Average Mode Subspace Overlap (MSO) across layers in the 65–90% depth range for pairwise comparisons of activations from Useful (U) and multiple Harmful (H1, H2) prompt sets.