DISENTANGLING SAFETY AND CAPABILITY IN LARGE LANGUAGE MODELS VIA CAUSAL INTERVENTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety alignment in large language models can deteriorate during downstream post-training because task learning and safety control remain coupled through shared Transformer computation. Existing safety-preserving methods primarily reduce geometric overlap between task- and safety-related parameter directions, but geometric separation does not directly characterize whether the resulting functions are independent. We introduce **LCID** (**L**inear-to-**C**ausal **I**nterventional **D**isentanglement), which decomposes post-training updates into shared, task-private, and safety-private components and uses controlled branch interventions to quantify their target and cross-functional effects. LCID directly suppresses task-to-safety and safety-to-utility cross-effects while preserving the intended contributions of each private branch and anchoring the complete model to its aligned safety behavior. Experiments across Qwen3-4B, Llama-3.1-8B, and GLM-4-9B show that LCID substantially improves the safety–utility trade-off and consistently reduces unsafe responses across multiple safety benchmarks. Prompt-level paired analysis further shows that these safety gains are systematic across harmful inputs. Intervention analysis demonstrates that near-orthogonal parameter updates can still exhibit substantial functional cross-talk, whereas LCID achieves markedly lower cross-effects while maintaining strong target effects. These results establish functional intervention as a direct criterion for disentangling safety and capability during post-training.

1 INTRODUCTION

Large language model (LLM) safety alignment aims to constrain model behavior on harmful, dangerous, or otherwise inappropriate requests, enabling responses such as refusal and safe redirection. Modern alignment pipelines combine supervised instruction tuning with preference-based optimization to shape helpful and safe behavior (Ouyang et al., 2022). Despite effective alignment at deployment time, downstream post-training can substantially degrade this behavior, including adaptation on benign task data (Qi et al., 2024). This phenomenon raises a central mechanistic question: how are safety and general capability coupled inside the model, and how can post-training improve one function without disrupting the other?

Prior work has identified low-dimensional refusal and safety-related structures in activation and parameter spaces (Arditi et al., 2024), and safety-preserving methods such as Safe LoRA and GuardSpace protect these structures through projection, orthogonalization, and null-space constraints (Hsu et al., 2024; Zhang et al., 2026). Recent mechanistic studies show that aligned behavior can be concentrated in compact activation directions (Arditi et al., 2024), while deeper token-level analyses reveal that safety alignment can remain shallow and vulnerable to subsequent optimization (Qi et al., 2025). Recent evidence further shows that safety-related and general-purpose learning are not cleanly separated in linear subspaces (Ponkshe et al., 2026). We build on this observation and distinguish *linear separation* from *functional independence*: the former concerns parameter geometry, whereas the latter concerns how activating one learned component changes another evaluated function. Figure 1 summarizes the central empirical motivation of this paper: capability gains remain behaviorally coupled to safety loss, linear protection only marginally shifts the operating point, and high linear decodability of risk does not guarantee effective behavioral control. Section 2 reviews these lines of work in detail.

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.

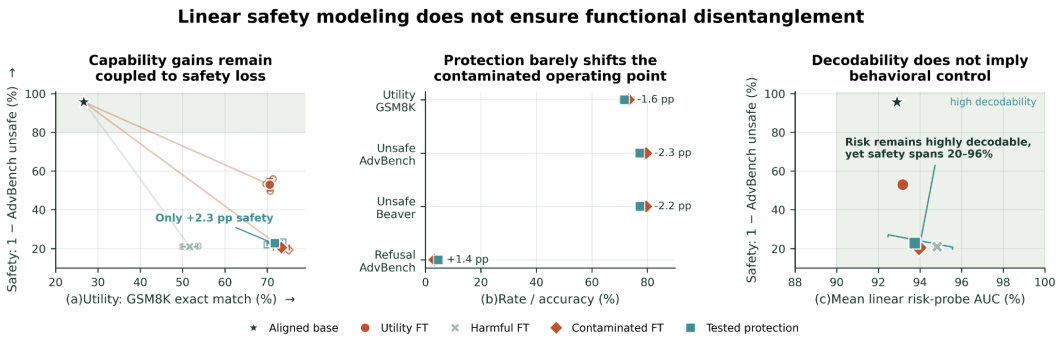

Figure 1: **Linear safety modeling does not ensure functional disentanglement.** **(a)** Capability improvements remain coupled to safety degradation under standard post-training, indicating persistent behavioral entanglement. **(b)** Linear subspace protection yields only limited safety recovery while slightly reducing utility, leaving the model near a contaminated safety–utility operating region. **(c)** Risk information remains highly linearly decodable even when behavioral safety varies substantially, showing that the presence of a decodable safety-related representation does not by itself imply behavioral control.

To model this distinction directly, we propose **LCID** (*Linear-to-Causal Interventional Disentanglement*). LCID decomposes post-training updates into shared, task-private, and safety-private components, making the task and safety branches independently switchable within the same checkpoint. Counterfactual comparisons between the complete model and branch-removed states quantify the task branch's effect on safety and the safety branch's effect on task utility. These effects are organized into a safety–capability intervention matrix and optimized through counterfactual functional invariance, target-effect preservation, and an aligned-model safety anchor.

Our experiments evaluate this functional criterion at behavioral, geometric, and intervention levels. LCID preserves downstream capability while sharply reducing harmful-response rates across GSM8K, MBPP, and IFEval post-training. Explicit orthogonalization drives parameter cosine similarity close to zero yet leaves substantially larger task-to-safety and safety-to-utility effects than LCID, directly separating geometric orthogonality from functional independence. Prompt-level paired analysis further shows large and statistically significant safety improvements across multiple model families. Ablations, controlled branch removal, and cross-backbone experiments connect these behavioral gains to reduced cross-functional effects and preserved target effects.

Our contributions are threefold. **(1)** We formulate safety–capability entanglement as measurable bidirectional intervention effects rather than parameter overlap alone. **(2)** We introduce LCID, an intervenable shared–private parameterization that directly optimizes functional independence while preserving useful task and safety effects. **(3)** We provide controlled evidence that stronger linear separation does not necessarily produce stronger functional separation, and show that LCID improves the safety–utility trade-off by directly minimizing cross-functional effects.

2 Related Work

Interpretability of safety mechanisms. Mechanistic analyses of language models have identified compact internal structures that encode behaviorally relevant information. Linear probing can recover latent knowledge from hidden activations (Burns et al., 2023), while Arditi et al. (2024) show that refusal can be modulated through a single residual-stream direction. Activation steering further demonstrates that targeted directions can causally modify model behavior at inference time (Li et al., 2023; Rimsky et al., 2024). Ponkshe et al. (2026) extend this geometric perspective to safety and show substantial overlap between safety-related and general-purpose learning in both weight and activation spaces. Together, these results motivate separating the existence of a linearly identifiable safety representation from the functional role that representation plays after post-training.

Safety preservation during fine-tuning. Safety-aligned models can lose refusal behavior after downstream fine-tuning, including adaptation on benign data (Qi et al., 2024). Token-level analyses further connect this fragility to shallow safety alignment and show that deeper alignment improves persistence under fine-tuning (Qi et al., 2025). Safety evaluation and red teaming provide complementary behavioral tests for such failures (Perez et al., 2022). A major line of work addresses this problem by protecting safety-related directions during optimization. Safe LoRA projects adapter updates toward a safety-aligned subspace (Hsu et al., 2024), while GuardSpace constructs safety-sensitive and harmful-resistant subspaces and constrains task updates through null-space projection (Zhang et al., 2026). These methods treat geometric relations among updates as the main control variable. LCID complements this perspective by directly measuring how task and safety components change each other's evaluated functions after propagation through the shared Transformer.

Intervention-based mechanistic analysis. Intervention methods provide a direct route from internal model components to behavioral effects. Causal tracing and model editing identify influential internal states by actively modifying them and measuring downstream changes (Meng et al., 2022). Complementary work extracts compact function vectors with causal effects on in-context behavior (Todd et al., 2024), while inference-time activation interventions steer truthfulness and other high-level behaviors (Li et al., 2023; Rimsky et al., 2024). LCID brings this intervention principle to safety-preserving post-training: task-private and safety-private parameter components are independently activated or removed within the same checkpoint, yielding explicit task-to-safety, safety-to-utility, and target effects. This formulation turns safety–capability disentanglement into an intervention-defined optimization problem rather than a geometric separation objective alone.

3 METHODS

LCID is motivated by the observation that parameter separation does not necessarily imply functional independence. As illustrated in Figure 2, even when task-related and safety-related updates are nearly orthogonal in parameter space, they are still propagated through the same attention, MLP, and residual pathways of the Transformer, and can therefore interfere with each other at the functional level. LCID addresses this problem in three stages. First, it decomposes post-training parameters into a shared component, a task-private component, and a safety-private component, making the two private components independently intervenable. Second, it constructs counterfactual model states by selectively activating or removing these components and measures four task–safety intervention effects. Third, it suppresses undesirable cross-effects while preserving the intended task and safety effects and anchoring the complete model to the original safety-aligned behavior.

3.1 FROM LINEAR SEPARATION TO FUNCTIONAL DISENTANGLEMENT

Let $F_A = f_{\theta_a}$ denote the original safety-aligned language model, and let $\mathcal{D}_U$ and $\mathcal{D}_S$ denote the utility-oriented and safety-oriented data distributions, respectively. A sample $x_u \sim \mathcal{D}_U$ corresponds to a downstream capability objective such as mathematical reasoning, code generation, or instruction following, whereas $x_s \sim \mathcal{D}_S$ corresponds to a safety objective associated with harmful or risky requests. Post-training aims to improve utility while preserving the safety behavior already established in F_A.

Existing safety-preserving methods commonly describe the interaction between the two objectives through parameter geometry. For example, task and safety updates can be encouraged to satisfy $\langle \Delta\theta_t, \Delta\theta_s \rangle \approx 0$. Such a constraint reduces their linear overlap, but it does not directly describe what happens after the two updates are propagated through the same Transformer. Two parameter components can be orthogonal while still changing the same hidden representations, attention patterns, or output distributions. We therefore distinguish *parameter separation* from *functional disentanglement*. Instead of asking only whether $\Delta\theta_t$ and $\Delta\theta_s$ occupy different directions, LCID asks two functional questions for each component: whether it improves its intended function, and whether it unintentionally changes the other function.

This gives four intervention effects. The task-private component may affect utility or safety, yielding $\mathcal{E}_{T\to U}$ and $\mathcal{E}_{T\to S}$; the safety-private component may likewise affect utility or safety, yielding $\mathcal{E}_{S\to U}$ and $\mathcal{E}_{S\to S}$. The task-to-utility and safety-to-safety effects are the desired *target effects*, while task-

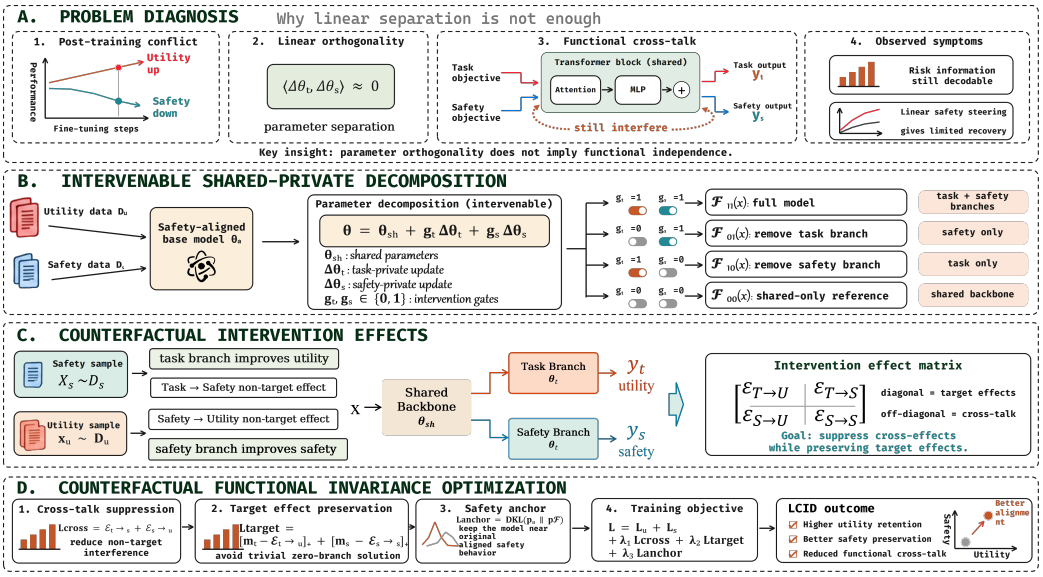

Figure 2: **Overview of LCID. A.** Although task and safety updates can be nearly orthogonal in parameter space, they still interact through the shared nonlinear Transformer computation, producing functional cross-talk. **B.** LCID decomposes the safety-aligned model into shared, task-private, and safety-private components controlled by intervention gates g_t and g_s, producing four counterfactual states F_{11}, F_{01}, F_{10}, and F_{00}. **C.** Counterfactual comparisons quantify four intervention effects: task-to-utility, task-to-safety, safety-to-utility, and safety-to-safety. The diagonal terms represent target effects, whereas the off-diagonal terms represent cross-functional interference. **D.** LCID minimizes cross-effects while preserving target effects and the safety behavior of the original aligned model.

to-safety and safety-to-utility are the undesired *cross-effects*. Functional disentanglement therefore corresponds to preserving strong diagonal effects while suppressing the off-diagonal effects.

3.2 Intervenable Shared–Private Decomposition

To measure the effect of a component, that component must first be independently controllable. LCID therefore rewrites the post-training parameters into a shared component and two private updates:

$$\begin{aligned} \theta(g_t, g_s) &= \theta_{\text{sh}} + g_t \Delta\theta_t + g_s \Delta\theta_s, \\ F_{g_t,g_s}(x) &= f_{\theta(g_t,g_s)}(x), \qquad g_t, g_s \in \{0,1\}. \end{aligned} \tag{1}$$

Here, θ_{sh} denotes the shared parameters used by both objectives, $\Delta\theta_t$ denotes the task-private update, and $\Delta\theta_s$ denotes the safety-private update. The binary variables g_t and g_s act as intervention gates: setting a gate to 1 activates the corresponding private component, while setting it to 0 removes that component from the current forward pass.

The decomposition is initialized from the safety-aligned model, with $\theta_{\text{sh}} \leftarrow \theta_a$ and the two private updates initialized from zero. Utility-oriented training primarily updates θ_{sh} and $\Delta\theta_t$, whereas safety-oriented training primarily updates θ_{sh} and $\Delta\theta_s$. The shared component therefore captures information useful to both objectives, while the private components provide explicit intervention variables for functional analysis.

The two gates produce four model states from the same learned checkpoint. F_{11} activates both task and safety branches and represents the complete model; F_{01} removes the task branch and therefore contains the shared backbone plus the safety branch; F_{10} removes the safety branch and contains the shared backbone plus the task branch; and F_{00} contains only the shared backbone. Importantly, these states do not require training four independent models. They share the same learned parameters and differ only in the branch selected by (g_t, g_s).

This construction provides the basis for counterfactual intervention. When comparing F_{11} with F_{01}, the shared parameters and safety branch remain fixed and only the task branch changes. Their difference can therefore be attributed to the intervention on $\Delta\theta_t$. Similarly, comparing F_{11} with F_{10} isolates the intervention on $\Delta\theta_s$. LCID then evaluates each of these two interventions on both $\mathcal{D}_U$ and $\mathcal{D}_S$, producing the four functional effects shown in Figure 3.

3.3 Counterfactual Intervention Effects

Figure 3 expands the intervention mechanism in Figure 2. The notation $\mathcal{E}_{A\to B}$ follows a simple rule: the symbol on the left of the arrow indicates *which private branch is intervened on*, while the symbol on the right indicates *which function is evaluated*. Consequently, the same task intervention $F_{11} \leftrightarrow F_{01}$ yields $\mathcal{E}_{T\to U}$ when evaluated on utility data and $\mathcal{E}_{T\to S}$ when evaluated on safety data. Similarly, the safety intervention $F_{11} \leftrightarrow F_{10}$ yields $\mathcal{E}_{S\to U}$ on utility data and $\mathcal{E}_{S\to S}$ on safety data.

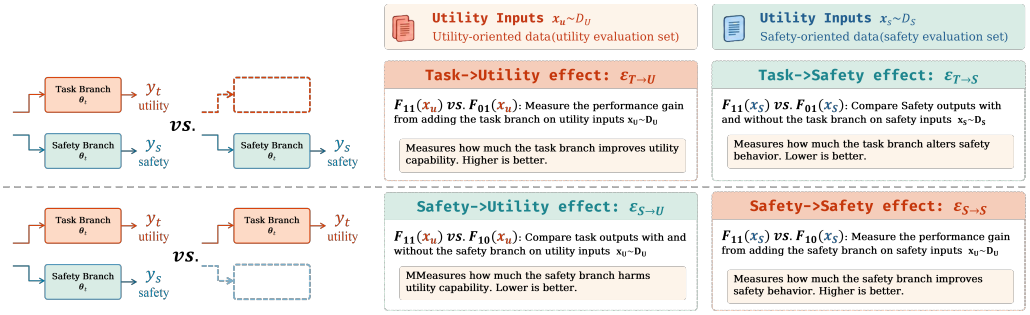

Figure 3: **Definition of the four counterfactual intervention effects.** The task intervention compares F_{11} and F_{01}, while the safety intervention compares F_{11} and F_{10}. Evaluating the same intervention on $\mathcal{D}_U$ or $\mathcal{D}_S$ determines whether it is a utility effect or a safety effect. $\mathcal{E}_{T\to U}$ and $\mathcal{E}_{S\to S}$ are target effects and should remain large, whereas $\mathcal{E}_{T\to S}$ and $\mathcal{E}_{S\to U}$ are cross-effects and should be minimized.

The four effects are defined jointly as

$$\begin{aligned}
\mathcal{E}_{T\to U} &= \mathbb{E}_{(x_u,y_u)\sim\mathcal{D}_U}\left[\ell_U(F_{01}(x_u), y_u) - \ell_U(F_{11}(x_u), y_u)\right], \\
\mathcal{E}_{T\to S} &= \mathbb{E}_{x_s\sim\mathcal{D}_S}\left[\mathrm{DL}(p_{F_{11}}(\cdot \mid x_s), p_{F_{01}}(\cdot \mid x_s))\right], \\
\mathcal{E}_{S\to U} &= \mathbb{E}_{x_u\sim\mathcal{D}_U}\left[\mathrm{DL}(p_{F_{11}}(\cdot \mid x_u), p_{F_{10}}(\cdot \mid x_u))\right], \\
\mathcal{E}_{S\to S} &= \mathbb{E}_{(x_s,y_s)\sim\mathcal{D}_S}\left[\ell_S(F_{10}(x_s), y_s) - \ell_S(F_{11}(x_s), y_s)\right].
\end{aligned} \tag{2}$$

Although all four quantities are derived from the same branch-removal principle, the target and cross-effects use different measurements because they answer different questions. For the target effects, LCID needs to determine whether activating the corresponding branch actually *improves* its intended objective. Therefore, $\mathcal{E}_{T\to U}$ and $\mathcal{E}_{S\to S}$ are defined as reductions in the supervised losses ℓ_U and ℓ_S. For example, $\mathcal{E}_{T\to U}$ compares the utility loss without the task branch, F_{01}, against the loss of the complete model, F_{11}. If the task branch contributes positively to utility, then $\ell_U(F_{11}) < \ell_U(F_{01})$ and $\mathcal{E}_{T\to U} > 0$. The same reasoning gives $\mathcal{E}_{S\to S} > 0$ when the safety branch improves safety.

The two cross-effects answer a different question: whether activating one branch changes the output associated with the other function. Here the direction of the change is less important than its magnitude, so LCID measures the discrepancy between the corresponding output distributions. $p_{F_{ij}}(\cdot \mid x)$ denotes the token probability distribution produced by state F_{ij}, and $\mathrm{DL}(\cdot,\cdot)$ denotes the distribution-level discrepancy used to quantify behavioral change. In our implementation, DL can be instantiated by KL divergence, Jensen–Shannon divergence, or token-level logit distance. Thus, a small $\mathcal{E}_{T\to S}$ means that activating the task branch barely changes the model's distribution on safety inputs, while a small $\mathcal{E}_{S\to U}$ means that activating the safety branch has little influence on utility predictions.

The four quantities naturally form the intervention effect matrix

$$\mathbf{E} = \begin{bmatrix} \mathcal{E}_{T\to U} & \mathcal{E}_{T\to S} \\ \mathcal{E}_{S\to U} & \mathcal{E}_{S\to S} \end{bmatrix}. \tag{3}$$

The rows correspond to the intervened branch and the columns correspond to the evaluated function. The diagonal terms are therefore target effects, whereas the off-diagonal terms quantify functional cross-talk. In this form, functional disentanglement has a direct interpretation: LCID seeks a nearly diagonal intervention matrix in which each private branch retains a strong effect on its intended function while exerting little influence on the other function.

3.4 Counterfactual Functional Invariance Optimization

The intervention matrix directly determines the training objective shown in Figure 2D. LCID simultaneously imposes three requirements: the off-diagonal effects should be small, the diagonal effects should remain sufficiently strong, and the complete model should remain close to the original safety-aligned behavior. We implement these requirements with the cross-talk loss, target-effect preservation loss, and safety anchor:

$$\begin{aligned} \mathcal{L}_{\text{cross}} &= \mathcal{E}_{T\to S} + \mathcal{E}_{S\to U}, \\ \mathcal{L}_{\text{target}} &= [m_t - \mathcal{E}_{T\to U}]_+ + [m_s - \mathcal{E}_{S\to S}]_+, \\ \mathcal{L}_{\text{anchor}} &= \mathbb{E}_{x_s\sim\mathcal{D}_S}\left[D_{\text{KL}}(p_{F_A}(\cdot \mid x_s) \,\|\, p_{F_{11}}(\cdot \mid x_s))\right], \\ \mathcal{L} &= \mathcal{L}_U + \mathcal{L}_S + \lambda_1\mathcal{L}_{\text{cross}} + \lambda_2\mathcal{L}_{\text{target}} + \lambda_3\mathcal{L}_{\text{anchor}}. \end{aligned} \tag{4}$$

$\mathcal{L}_{\text{cross}}$ directly minimizes the two off-diagonal entries of $\mathbf{E}$. On safety inputs, the task-private branch is therefore trained to leave safety behavior invariant; on utility inputs, the safety-private branch is trained to leave task behavior invariant. This is the *counterfactual functional invariance* principle of LCID: removing a non-target branch should produce little change in the function that the branch is not intended to control.

Minimizing cross-effects alone could reduce the influence of the private branches themselves. LCID therefore retains their intended contributions through $\mathcal{L}_{\text{target}}$. The margins m_t and m_s specify the minimum desired task-to-utility and safety-to-safety effects, respectively, and $[z]_+ = \max(0, z)$. If $\mathcal{E}_{T\to U}$ already exceeds m_t, the task branch receives no additional target-effect penalty; the same applies to the safety branch once $\mathcal{E}_{S\to S} \geq m_s$. The target loss therefore preserves functional specialization while the cross loss removes functional leakage.

The safety anchor provides a fixed behavioral reference throughout post-training. $p_{F_A}(\cdot \mid x_s)$ denotes the output distribution of the original safety-aligned model, whereas $p_{F_{11}}(\cdot \mid x_s)$ denotes that of the complete LCID model. Minimizing $\mathcal{L}_{\text{anchor}}$ keeps the learned model close to the original aligned behavior on safety-oriented inputs while the task and safety branches are being reorganized.

Finally, $\mathcal{L}_U$ and $\mathcal{L}_S$ are the ordinary supervised objectives on $\mathcal{D}_U$ and $\mathcal{D}_S$, and λ_1, λ_2, and λ_3 control the relative strengths of cross-effect suppression, target-effect preservation, and safety anchoring. The complete objective therefore has a direct functional interpretation: $\mathcal{L}_U + \mathcal{L}_S$ determines the behaviors the model should learn, $\mathcal{L}_{\text{target}}$ ensures that the corresponding private branches remain effective, $\mathcal{L}_{\text{cross}}$ prevents those branches from interfering with the other function, and $\mathcal{L}_{\text{anchor}}$ preserves the original safety reference.

Through this construction, LCID turns safety–utility disentanglement into an explicit intervention problem. Parameter decomposition provides controllable variables, branch removal provides matched counterfactual comparisons, the intervention matrix quantifies target and cross-functional effects, and the final objective directly shapes that matrix toward strong diagonal effects and weak off-diagonal interference.

4 Experiments and Results

We evaluate LCID from three complementary perspectives: **(1)** whether it improves the safety–utility trade-off during post-training, **(2)** whether the resulting safety improvements are consistent at the prompt level, and **(3)** whether functional disentanglement can be distinguished from parameter-space orthogonality. Additional ablations and branch-level counterfactual analyses are reported in the appendix.

4.1 EXPERIMENTAL SETUP

Models and evaluation. Qwen3-4B (Qwen Team, 2025) is used as the primary backbone, with Llama-3.1-8B and GLM-4-9B included for cross-model evaluation. Utility is evaluated on GSM8K (Cobbe et al., 2021), MBPP (Austin et al., 2021), and IFEval (Zhou et al., 2023), covering mathematical reasoning, code generation, and instruction following, respectively. Safety is evaluated using unsafe-response rate (UR) on AdvBench (Zou et al., 2023), BeaverTails-H (Ji et al., 2023), and StrongREJECT (Souly et al., 2024). We additionally report benign refusal to measure unnecessary refusal on harmless inputs.

Baselines. We compare LCID with the aligned base model, standard fine-tuning, SafeLoRA, GuardSpace, Shared–Private decomposition, and Shared–Private + Orth. The latter two baselines isolate the effect of structural parameter decomposition and explicit geometric separation from LCID's intervention-based functional objective.

Paired behavioral analysis. To complement aggregate unsafe-response rates, we perform paired analysis on 300 BeaverTails-H prompts for each evaluated backbone. We report paired-bootstrap confidence intervals and use exact two-sided McNemar tests with Holm correction to characterize the direction and consistency of prompt-level behavioral changes.

4.2 MAIN SAFETY–UTILITY RESULTS

Table 1 compares utility and safety across model backbones. Standard fine-tuning produces strong downstream capability but substantially degrades safety, illustrating the central conflict between task adaptation and previously aligned behavior. Safety-preserving baselines partially recover the lost safety, but typically move the model to a different point on the safety–utility trade-off rather than eliminating the interaction between the two objectives.

Table 1: **Main safety–utility results across model backbones.** Utility is evaluated on GSM8K, MBPP, and IFEval. Safety is measured by unsafe-response rate (UR) on AdvBench, BeaverTails-H, and StrongREJECT (SR). Utility Avg. and Safety UR Avg. denote the mean performance across the corresponding three benchmarks. Benign Ref. denotes the refusal rate on benign prompts. ↑ and ↓ indicate higher and lower is better, respectively.

Backbone	Method	Utility (%) ↑				Safety: Unsafe Rate (%) ↓				Over-refusal
		GSM8K	MBPP	IFEval	Avg.	AdvBench	Beaver-H	SR	Avg.	Benign Ref. ↓
Qwen3-4B	Aligned Base	27.67	41.80	64.90	44.79	4.00	6.20	3.50	4.57	18.40
	Standard FT	**72.02**	**72.80**	78.50	**74.44**	31.83	48.53	28.70	36.35	**5.60**
	SafeLoRA	64.80	68.20	75.50	69.50	13.70	12.87	10.10	12.22	14.20
	GuardSpace	67.20	69.60	77.00	71.27	7.93	7.47	10.10	8.50	15.30
	Shared–Private	69.50	71.30	78.40	73.07	19.80	17.53	13.63	16.99	10.70
	Shared–Private + Orth.	68.60	70.50	77.90	72.33	12.70	10.87	9.20	10.92	12.10
	LCID (Ours)	**72.02**	61.80	**80.34**	71.39	**5.33**	**5.33**	**1.60**	**4.09**	18.00
Llama-3.1-8B	Orth.	69.04	55.07	72.03	65.38	78.00	78.00	60.06	72.02	**2.33**
	LCID (Ours)	**71.06**	**57.04**	**75.02**	**67.71**	**10.33**	**9.33**	**3.08**	**7.58**	10.00
GLM-4-9B	Orth.	67.05	56.02	70.07	64.38	77.67	78.33	60.04	72.01	**3.67**
	LCID (Ours)	**69.53**	**59.06**	**73.02**	**67.20**	**21.00**	**20.67**	**6.54**	**16.07**	9.67

LCID produces a different pattern. On Qwen3-4B, it maintains strong mathematical reasoning and instruction-following performance while achieving the lowest average unsafe-response rate among the post-training methods. Compared with the Shared–Private and orthogonalized variants, LCID provides substantially stronger safety without requiring the most geometrically separated parameter updates. Its benign-refusal behavior also remains close to that of the original aligned model, indicating that the recovered safety behavior is not obtained simply by shifting the model toward indiscriminate refusal.

The same trend is reproduced on Llama-3.1-8B and GLM-4-9B. Across both backbones, LCID improves downstream utility relative to parameter orthogonalization while sharply reducing unsafe-

response rates. The consistency across model families suggests that the benefit of LCID arises from the functional intervention objective rather than a backbone-specific parameter configuration.

4.3 Prompt-Level Consistency of Safety Improvement

The aggregate results in Table 1 describe the overall operating point, while Figure 4 examines how that improvement is distributed across individual prompts.

LCID shows a large and consistently positive unsafe-rate reduction relative to standard fine-tuning across all three backbones, with paired-bootstrap confidence intervals clearly separated from zero. Linear orthogonalization, in contrast, produces only small paired changes around the original operating point.

The transition analysis provides the same conclusion from a discrete behavioral perspective. For LCID, unsafe-to-safe transitions strongly dominate the reverse direction across model families, whereas orthogonalization produces approximately balanced transitions. The agreement between the bootstrap effects and McNemar tests shows that LCID's aggregate safety improvement reflects systematic prompt-level behavioral correction.

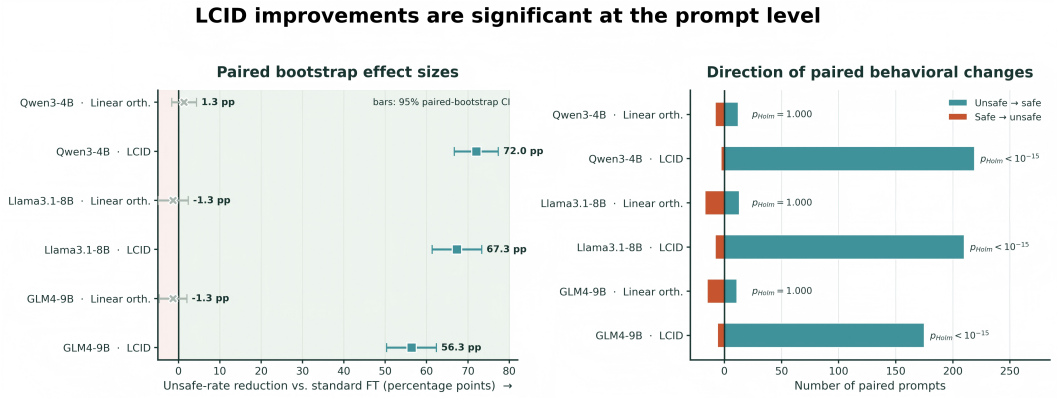

Figure 4: **Prompt-level significance analysis.** Left: paired-bootstrap unsafe-rate reductions relative to standard fine-tuning with 95% confidence intervals. Right: paired unsafe-to-safe and safe-to-unsafe transitions on BeaverTails-H, evaluated using exact two-sided McNemar tests with Holm correction. LCID produces consistently directed safety improvements across model families, whereas linear orthogonalization yields substantially weaker behavioral changes.

4.4 Geometric Separation versus Functional Disentanglement

Table 2 provides the central comparison between parameter geometry and functional intervention effects. Adding an explicit orthogonal constraint almost eliminates cosine overlap between the task-private and safety-private updates. However, substantial task-to-safety and safety-to-utility effects remain after this geometric separation.

LCID exhibits the opposite pattern: its parameter updates are less orthogonal, yet its two cross-effects are substantially smaller. At the same time, LCID retains the strongest task-to-utility and safety-to-safety target effects. The resulting intervention matrix is therefore more strongly diagonal: each private component has a larger influence on its intended function and a smaller influence on the other function.

This comparison directly separates the two notions of disentanglement. Parameter orthogonality describes the geometric relationship between updates, whereas the intervention effects measure what those updates actually change after passing through the shared nonlinear Transformer. The results show that stronger geometric separation does not necessarily produce stronger functional separation.

Figure 5 connects this distinction to the observed model behavior. The cross-model trajectories show that LCID consistently moves post-trained models toward a safer operating region while retaining

Table 2: **Parameter geometry versus functional intervention effects.** Orthogonalization nearly eliminates parameter-space overlap, whereas LCID achieves substantially lower cross-functional effects while preserving stronger target effects. ↓ and ↑ indicate lower and higher is better, respectively.

Method	Geometry	Cross-effects		Target-effects	
	$\lvert\cos(\Delta\theta_t, \Delta\theta_s)\rvert \downarrow$	$\mathcal{E}_{T\to S} \downarrow$	$\mathcal{E}_{S\to U} \downarrow$	$\mathcal{E}_{T\to U} \uparrow$	$\mathcal{E}_{S\to S} \uparrow$
Shared–Private	0.312	0.186	0.142	<u>0.284</u>	<u>0.261</u>
+ Orthogonal Constraint	**0.008**	<u>0.121</u>	<u>0.097</u>	0.273	0.258
LCID (Ours)	<u>0.094</u>	**0.031**	**0.027**	**0.291**	**0.276**

utility. The middle panel shows that this behavioral recovery persists even when linear orthogonalization itself provides little safety improvement. Finally, the counterfactual analysis shows that the learned branches exhibit functional selectivity: their target effects dominate the corresponding cross-effects. Together, these results connect the improved safety–utility trade-off to intervention-defined specialization rather than parameter geometry alone.

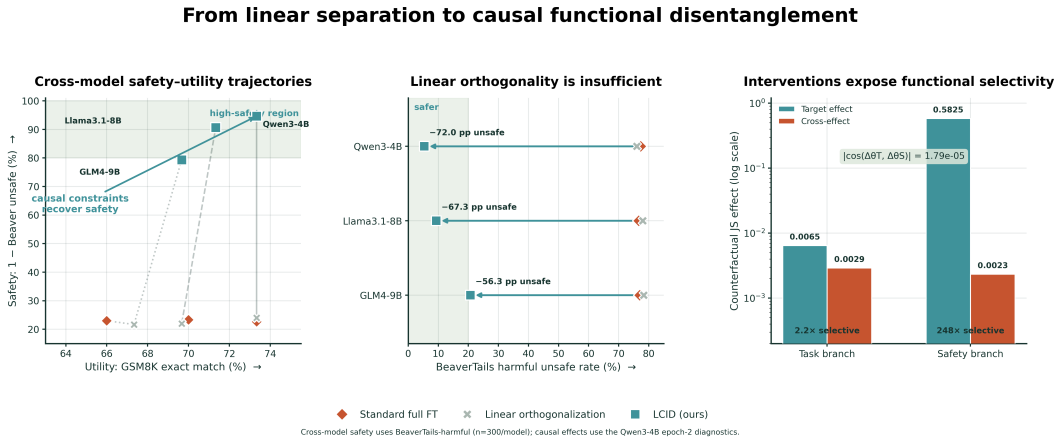

Figure 5: **From linear separation to causal functional disentanglement.** Left: cross-model safety–utility trajectories under standard fine-tuning, linear orthogonalization, and LCID. Middle: changes in BeaverTails-H unsafe-response rate from linear orthogonalization to LCID. Right: counterfactual target and cross-effects illustrating branch-level functional selectivity. Together, the three views connect behavioral safety recovery to functional intervention rather than parameter orthogonality alone.

5 Conclusion

LCID reframes safety–capability disentanglement as a functional intervention problem. By combining an intervenable shared–private parameterization with counterfactual functional invariance, target-effect preservation, and safety anchoring, LCID directly suppresses task-to-safety and safety-to-utility cross-effects while retaining useful target contributions. Experiments show that near-orthogonal parameter updates can remain functionally coupled, whereas LCID substantially reduces cross-effects, improves the safety–utility trade-off across downstream tasks, and preserves the same pattern across model backbones. These results establish functional intervention as a direct criterion for disentangling safety and capability during post-training.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and manuscript editing. All technical claims, equations, experimental results, citations, and final text were reviewed by the authors, who take 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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html.

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

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

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

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/77baa7c2a3a675823e89131698fd6e19-Abstract-Conference.html.

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. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract.html. Datasets and Benchmarks Track.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.225. URL https://aclanthology.org/2022.emnlp-main.225/.

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*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9a9f4e15ad0d680429a3e0570a96f763-Abstract-Conference.html.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

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. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/88be023075a5a3ff3dc3b5d26623fa22-Abstract-Conference.html.

Qwen Team. Qwen3-4B model card. Hugging Face model repository, 2025. URL https://huggingface.co/Qwen/Qwen3-4B.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e2e06adf560b0706d3b1ddfca9f29756-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4ae163cb8788970e53b4fd9578141139-Abstract-Conference.html.

Bingjie Zhang, Yibo Yang, Zhe Ren, Dandan Guo, Jindong Gu, Philip Torr, and Bernard Ghanem. A guardrail for safety preservation: When safety-sensitive subspace meets harmful-resistant null-space. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d949ad2d45ba557f77327a5f720e73d3-Abstract-Conference.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

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. URL https://arxiv.org/abs/2307.15043.