CAUSAL FRONT-DOOR ADJUSTMENT FOR ROBUST JAILBREAK ATTACKS ON LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Jailbreak attacks on safety-aligned LLMs are increasingly effective, yet attack success alone does not reveal whether compliance is driven by preserved task intent, suppressed safety signals, or attack-specific artifacts. We use Pearl's front-door adjustment to motivate separating task-relevant semantics from safety-related variation. We propose **Causal Front-Door Adjustment Attack** (**CFA**^{2}), a white-box framework that operationalizes this perspective in LLM latent space using Sparse Autoencoders, contrastive datasets, and weight orthogonalization. After offline calibration, CFA^2 enables efficient generation without per-prompt iterative optimization, performs strongly on HarmBench across four open-source LLMs, and provides source-side signals for same-family black-box prompt transfer.

1 INTRODUCTION

Large Language Models (LLMs) Qi et al. (2024); Chao et al. (2024); Liu et al. (2024a) have revolutionized human-computer interaction, yet they expose significant safety vulnerabilities. Even models aligned with rigorous safety protocols can be manipulated to generate harmful content through jailbreaking attacks Zhang & Wei (2025); Liu et al. (2024b); Li et al. (2023).

Recent jailbreak methods have demonstrated strong effectiveness through diverse intervention interfaces, including adversarial suffix optimization, semantic prompt rewriting, output-space manipulation, and representation-level steering Zou et al. (2023b); Chao et al. (2024); Liu et al. (2024a). However, high attack success does not by itself explain why a model complies. Since many methods operate by adding adversarial suffixes, role-playing contexts, output prefixes, or other auxiliary signals, the resulting input differs from the original harmful request in more than one aspect. Thus, attack success alone cannot reveal whether compliance is driven by preserved task intent, suppressed safety signals, or other reasons.

Motivated by this challenge, we investigate the underlying generation mechanism of LLMs through a causal lens. Our approach leverages the insight that safety alignment may act as a superficial constraint rather than erasing underlying knowledge Zhou et al. (2023); Wei et al. (2023a). Specifically, pre-trained models possess the capability to answer harmful queries, but this capability is suppressed by a latent safety mechanism introduced during alignment. In causal terms, jailbreak failures occur not because the model "cannot" answer, but because the safety mechanism acts as a confounder that interferes with the output.

Building on this insight, we formulate a Structural Causal Model (SCM), as illustrated in Figure 1. Let X denote the harmful query and Y the response. We introduce an unobserved node U to represent the internal safety mechanism. Crucially, we model U as a latent confounder: it influences how the query is processed (path $U \rightarrow A$, A is the model's internal latent representation of X) and directly biases the output towards refusal ($U \rightarrow Y$). This confounding effect explains why simple optimization fails; that is, it cannot distinguish confounded associations between the model's true capability and the safety interference.

To account for this confounding structure, we leverage Pearl's Front-Door Adjustment Pearl (2009). We introduce an observable mediator S, representing the core task semantics of the query. We instantiate S as the task-relevant semantic component of the query, which is designed to mediate the query-response pathway while minimizing its dependence on safety-related variation. Under this construction, front-door adjustment provides a principled way to estimate the interventional effect

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.

of the task pathway despite the unobserved safety mechanism U. Finally, we propose the **C**ausal **F**ront-Door **A**djustment **A**ttack (**CFA**^{2}), a framework that translates this causal theory into a practical, efficient attack algorithm. Specifically, CFA^2 implements this framework via a two-stage pipeline: First, employing Sparse Autoencoders with contrastive analysis to extract interpretable features and separate approximately invariant task-related features from defense mechanisms; and second, applying weight orthogonalization to suppress the identified defense-related direction, yielding a fixed parameter-level intervention with $O(1)$ attack-specific overhead with respect to per-prompt optimization rounds, followed by standard autoregressive decoding.

Our key contributions: (1) We propose a **causal formulation of jailbreak intervention** that models safety-related mechanisms as latent factors entangled with task-relevant semantics. Rather than treating jailbreak success purely as an input-output phenomenon, our formulation characterizes the transition from refusal to compliance as a problem of separating the task pathway from safety-induced interference. (2) We introduce **CFA**^{2}, an offline-calibrated and inference-efficient jailbreak algorithm. CFA^2 uses SAEs to identify approximately invariant task-related features. Crucially, we employ *Weight Orthogonalization* to project out estimated defense-related directions. This yields $O(1)$ attack-specific overhead with respect to the number of per-prompt optimization steps. (3) We provide **extensive empirical evidence** across effectiveness, efficiency, and transferability. CFA^2 performs strongly on HarmBench across four open-source LLMs, outperforming prompt-level attacks as well as stronger white-box or internal-state baselines such as ORTHO, DSN, and SM. CFA^2 further enables standard autoregressive generation without per-prompt iterative optimization and provides useful source-side signals for same-family black-box prompt transfer.

2 Related Work

LLM Jailbreak. Existing jailbreak methodologies can be broadly grouped into three categories: optimization-based adversarial attacks Zou et al. (2023b); Liu et al. (2024a); Zhang & Wei (2025); Zhou et al. (2025); Jin et al. (2024), semantic manipulation Chao et al. (2024); Li et al. (2023); Mehrotra et al. (2024); Chen et al. (2024); Li et al. (2024); Shah et al. (2023); Yong et al. (2024); Deng et al. (2024); Yuan et al. (2024); Yao et al. (2024), and fine-tuning or decoding interventions Zhang et al. (2024a); Huang et al. (2023). These methods manipulate prompts, decoding procedures, or model internals to increase harmful compliance. Representation-level attacks are most relevant to our setting. Refusal Direction Orthogonalization (ORTHO) identifies a dense refusal-related direction Arditi et al. (2024). Don't Say No (DSN) suppresses refusal behavior through optimization-based intervention Zhou et al. (2025), while Internal Stance Manipulation (SM) modifies internal states to shift the model toward compliance Fu et al. (2025). Although effective, these methods typically identify refusal or behavior related directions without explicitly controlling for task-semantic variation. More broadly, many jailbreak methods alter multiple aspects of the original request or generation process. This limits their interpretability as red-teaming tools, since the observed compliance does not clearly indicate whether success arises from preserved task intent, suppressed refusal features, or attack-specific artifacts.

Representation Learning and Mechanistic Interpretability. Representation learning provides a useful framework for analyzing and manipulating internal LLM states. Representation Engineering (RepE), for example, steers model behavior through dense activation directions Zou et al. (2023a), dense representations can be limited by polysemanticity and feature superposition, where dense representations entangle multiple semantic factors Elhage et al. (2022). To address this, SAE utilizes high-dimensional sparse projections to resolve feature superposition. Recent work has used SAEs to identify localized features associated with semantic concepts and model behaviors Huben et al. (2023); Bricken et al. (2023); Templeton (2024). However, current SAE research is predominantly limited to descriptive analysis or benign steering (e.g., enhancing honesty) Subramani et al. (2022), leaving their potential for adversarial red-teaming comparatively unexplored.

3 Problem Formulation and Causal Analysis

In this section, we establish the theoretical foundation for our proposed method. We begin by defining the objective of jailbreak attacks and introducing the mathematical notations in Section 3.1. Building on this formulation, we then transition to a causal perspective in Section 3.2. Here, we construct a

Structural Causal Model (SCM) to characterize the interference of unobserved safety mechanisms and apply the Front-Door Adjustment to theoretically derive a robust jailbreaking strategy.

3.1 Problem Formulation

In this subsection, we formally define the objective of the LLM jailbreak attack. Modern LLMs operate under an inherent tension: they retain vast knowledge and instruction-following capabilities from pre-training, but are constrained by safety alignment. A **jailbreak attack** is an adversarial attempt to bypass these constraints, aiming to elicit harmful, unethical, or restricted responses.

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be a dataset of harmful requests drawn from a distribution $P(x)$. Given a pre-trained LLM π_θ, the model defines a conditional distribution $\pi_\theta(y \mid x)$ over the response space $\mathcal{Y}$. For any input x, the response y is sampled autoregressively, i.e., $y \sim \pi_\theta(\cdot \mid x)$. We further define a binary judge function $g : \mathcal{Y} \to \{0, 1\}$, which determines whether a response y is a successful jailbreak ($g(y) = 1$) or a harmless refusal ($g(y) = 0$).

The objective of a jailbreak attack is to maximize the expected success rate, formally defined as:

$$P_{\text{success}} \triangleq \mathbb{E}_{x\sim P(x)}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[\mathbb{1}[g(y) = 1]\big], \tag{1}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. Intuitively, this objective aims to find the optimal strategy (e.g., input perturbations) to maximize the likelihood of harmful outputs. This essentially attempts to shift the model's output distribution from the safe region toward the harmful space.

3.2 Causal Analysis

From the problem formulation, our objective is to jailbreak a safety-aligned language model. In principle, a pre-trained model possesses the inherent capability to generate practical responses to arbitrary queries, regardless of their safety status. However, safety alignment introduces a constraint where the model produces refusals or templated responses for specific inputs. Crucially, such refusals do not imply a lack of underlying capability; rather, they indicate the presence of unobserved internal mechanisms (i.e., safety guardrails) that actively suppress the answer. To understand this suppression and recover the model's answering capability, we model the interactions between queries, responses, and safety mechanisms through a causal lens.

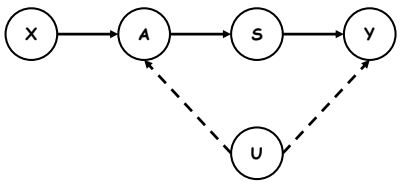

Figure 1: **Jailbreaking SCM.** X: query, A: representation, S: semantics, Y: harmful response, and U: latent safety mechanism.

The Structural Causal Model (SCM). We denote the input query by X, the response by Y, and the unobserved internal safety mechanism by U. Additionally, we introduce A to represent the model's internal latent representation of X.

In our causal graph (see Figure 1), the path $X \to A \to Y$ represents the model's inherent task capability at a coarse level, e.g., the ideal mapping from query to representation to response in the absence of constraints. However, safety alignment introduces a confounder U. The safety mechanism U influences the internal representation A (making observed representations a mixture of task and safety signals, $U \to A$) and explicitly biases the output Y towards refusal ($U \to Y$) Arditi et al. (2024); Wolf et al. (2024). Consequently, the observed conditional distribution $P(Y \mid X)$ is a mixture of the model's potential capability and the safety mechanism's intervention. To isolate the model's original answering capability, we introduce a conceptual mediator variable S, representing the core task semantics within X. We posit that the causal effect of task information on Y is strictly mediated through S, and we assume that S intercepts the directed effect of A on Y. Structurally, this decomposes the generation process into two stages: extracting the core task representation ($X \to A \to S$) and generating the response conditioned on this task-specific semantic ($S \to Y$). By introducing S, we isolate the mapping from X to Y driven solely by task information, effectively filtering out the non-task safety activations present in A.

Front-door Adjustment for Confounding Resolution. To formally characterize the genuine influence of the input on the response while shielding the process from the safety mechanism U, we employ the causal intervention operator $do(\cdot)$. The objective of jailbreak is defined as the

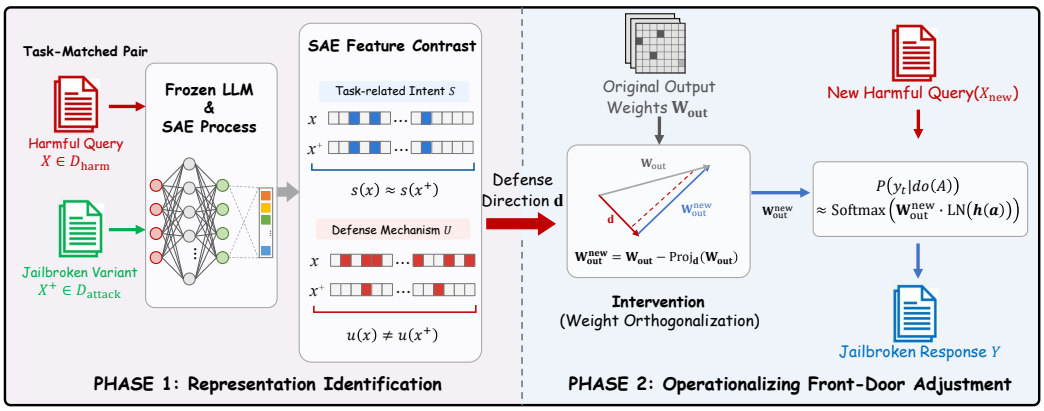

Figure 2: Overview of the Causal Front-Door Adjustment Attack (CFA^2) framework. CFA^2 uses task-matched harmful/jailbroken pairs to identify approximately invariant task-related features and a defense-related direction $\mathbf{d}$. It then projects $\mathbf{W}_{\text{out}}$ onto the orthogonal complement of $\mathbf{d}$, yielding modified weights $\mathbf{W}_{\text{out}}^{\text{new}}$. This parameter-level intervention suppresses the identified defense-related contribution while retaining task-related information, enabling standard autoregressive generation without per-prompt iterative optimization.

interventional distribution $P(Y \mid do(A))$. It represents the response distribution under an intervention that externally sets A, thereby removing the confounding influence on A.

Pearl's Front-door Criterion Pearl (2009) provides the theoretical foundation for identifying $P(Y \mid do(A))$ even when the confounder U is unobserved. Specifically, because the mediator S shields the direct influence of input features on Y, and the path from A to S is not intercepted by U, the causal intervention probability can be derived as:

$$P(Y = \boldsymbol{y}|do(A = \boldsymbol{a})) = \sum_s P(S = s|A = \boldsymbol{a}) \sum_{\boldsymbol{a}'} P(Y = \boldsymbol{y}|A = \boldsymbol{a}', S = s) P(A = \boldsymbol{a}') \quad (2)$$

Intuitively, the outer summation captures the activation of the task semantics S by the input $\boldsymbol{a}$. The inner expectation marginalizes over the input space $\boldsymbol{a}'$, thereby adjusting for the spurious association induced by the unobserved confounder U. This implies that we do not need to model the defense mechanism U explicitly; instead, we can use the mediator S to identify the task-mediated pathway through which the effect of $do(A)$ can be recovered.

Inspiration and Challenges. The front-door adjustment framework provides a causal view for separating task semantics from safety-related confounding. However, applying it to LLM jailbreak intervention requires addressing three challenges. First, **Representation Identification**: the mediator S should be located in high-dimensional, polysemantic activations. Second, **Estimation Efficiency**: the required marginalization over background representations A' is infeasible to compute directly. Third, **Operationalization**: the causal adjustment should be translated into a lightweight model-level intervention that preserves task semantics while suppressing safety-related variation.

4 Methodology

To address the aforementioned challenges, we propose a novel method termed **C**ausal **F**ront-Door **A**djustment **A**ttack (**CFA**^{2}). The central idea of CFA^2 is to leverage causal inference principles to design a causality-inspired jailbreak strategy for LLM. The method comprises two stages. In the first stage, we employ a Sparse Autoencoder (SAE) to decompose the LLM hidden representations into sparse latent features, and use task-matched contrastive analysis to identify approximately invariant task-related features and defense-related variation. In the second stage, we translate the identified defense-related variation into a residual-stream direction and apply weight orthogonalization to suppress its contribution to the output mapping. The resulting parameter-level intervention provides a tractable approximation to the marginalization effect motivated by the front-door formulation and is then used with standard autoregressive decoding.

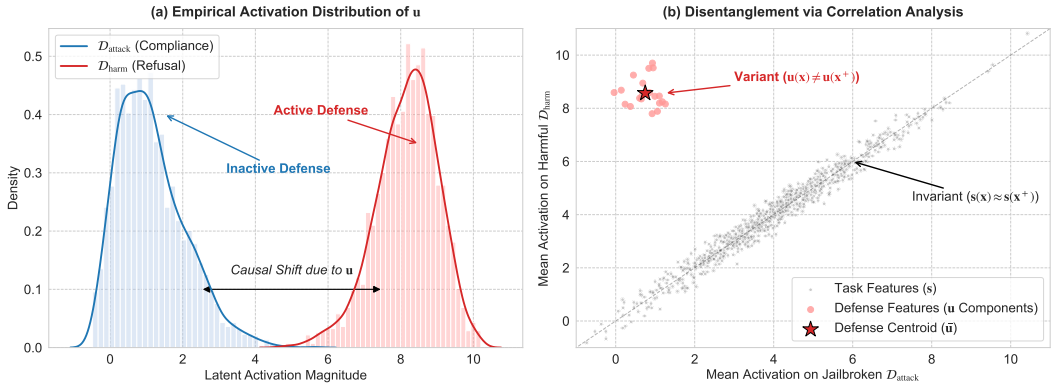

Figure 3: Validation of Disentanglement (Proposition 4.2). (a) Evidence of Style Variance: The activation distribution of $\boldsymbol{u}$ shows a distinct causal shift between the refusal state ($\mathcal{D}_{harm}$) and compliance state ($\mathcal{D}_{attack}$), confirming $\boldsymbol{u}(\boldsymbol{x}) \neq \boldsymbol{u}(\boldsymbol{x}^+)$. (b) Evidence of Content Invariance: While the defense mechanism (red star) acts as an outlier, the core task semantics (gray points, $\mathbf{s}$) remain aligned along the invariance line ($y = x$).

4.1 Identification of the Front-Door Mediator

A central challenge in implementing front-door adjustment is the accurate identification and formalization of the mediator S from observed data. In LLM tasks, the model's internal reasoning logic is highly compressed and encoded within dense, high-dimensional hidden activations Geva et al. (2021); Park et al. (2023); Meng et al. (2022). The inherent polysemanticity and superposition of these representations make it extremely challenging to extract the core task representation. To address this challenge, we draw inspiration from recent advances in SAE Ng et al. (2011); Ferrando et al. (2024), which have been shown to disentangle complex hidden activations into interpretable and localized sparse features. Specifically, an SAE is constructed as a neural network with an encoder function $\mathbf{z} = \phi(\mathbf{W}\mathbf{x} + \mathbf{b})$ and a decoder function $\hat{\mathbf{x}} = \mathbf{W}'\mathbf{z} + \mathbf{b}'$where $\mathbf{x} \in \mathbb{R}^d$ denotes the dense hidden activation vector of the LLM, $\mathbf{z} \in \mathbb{R}^k$ is the sparse latent representation, while $\mathbf{b} \in \mathbb{R}^k$ and $\mathbf{b}' \in \mathbb{R}^d$ correspond to the bias vectors, and ϕ are nonlinear activation functions. The training objective combines a reconstruction loss with a sparsity-inducing penalty, where λ controls the trade-off between accuracy and sparsity. The final optimization target is formulated as:

$$\min_{\mathbf{W},\mathbf{W}',\mathbf{b},\mathbf{b}'} \mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda\|\mathbf{z}\|_1 \tag{3}$$

Through this formulation, the autoencoder learns to reconstruct hidden activations while forcing most latent dimensions in $\mathbf{z}$ to remain inactive via L1 regularization. By applying an SAE to the hidden layers of an LLM, the dense activation latent vectors are projected into a sparse latent space, where each dimension corresponds to a candidate feature with clearer semantic interpretation. This transformation provides a principled means of searching for front-door variables: candidate variables are defined as the sparse latent dimensions extracted by the SAE, which can then be systematically examined to test whether they mediate the causal pathway between input prompts and model outputs. In this way, SAE serves not only as a dimensionality reduction tool but also as a bridge between raw model activations and causally meaningful variables.

To establish a theoretical basis for identifying the desired latent factors from SAE representations, we invoke the identifiability theory from the framework of nonlinear Independent Component Analysis (ICA) and disentangled representation learning Khemakhem et al. (2020); Von Kügelgen et al. (2021).

Proposition 4.1 (Identifiability of Latent Causal Factors)**.** *Let the data generating process be $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$, where $\boldsymbol{g} : \mathcal{Z} \to \mathcal{X}$ is a smooth diffeomorphism and $\boldsymbol{z} \in \mathbb{R}^{d_z}$ are latent factors. Assume the prior $p(\boldsymbol{z}|y)$ conditioned on the response y follows a conditionally factorial exponential family with sufficient statistics $\boldsymbol{T}(\boldsymbol{z})$ and parameters $\boldsymbol{\lambda}(y)$. If the learned representation $\boldsymbol{a} = \boldsymbol{f}(\boldsymbol{x})$ satisfies the following conditions:*

(1) ***Reconstruction Consistency****: The SAE reconstructs the input density, i.e., $p_{SAE}(\boldsymbol{x}) = p_{true}(\boldsymbol{x})$.*

(2) ***Task Sufficiency****: The SAE features preserve task information: $D_{KL}(p_{SAE}(y|\boldsymbol{x})\|p_{true}(y|\boldsymbol{x})) = 0$.*

*(3) **Variability**: The responses are diverse enough that $\boldsymbol{L} = [\boldsymbol{\lambda}(y_1) - \boldsymbol{\lambda}(y_0), \ldots, \boldsymbol{\lambda}(y_k) - \boldsymbol{\lambda}(y_0)]$ has rank k, the dimension of the sufficient statistics.*

*(4) **Sparsity Prior**: The true latent components z_i follow a super-Gaussian distribution, and the learning objective minimizes the ℓ_1-norm proxy for sparsity, i.e., $\min \mathbb{E}[\|\boldsymbol{a}\|_1]$.*

Then, the learned representation $\boldsymbol{a}$ identifies the true latent factors $\boldsymbol{z}$ up to a composition of a permutation matrix $\boldsymbol{P}$ and a diagonal scaling matrix $\boldsymbol{\Lambda}$. That is, $\boldsymbol{a} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{z} + \boldsymbol{c}$.

The proof is presented in Appendix B. While Proposition 4.1 establishes that the learned representation $\boldsymbol{a}$ is an identifiable linear transformation of the true latent factors $\boldsymbol{z} = [\boldsymbol{s}, \boldsymbol{u}]$, it does not explicitly disentangle the task-specific intent $\boldsymbol{s}$ from the defense mechanism $\boldsymbol{u}$ without supervision. Direct identification of $\boldsymbol{s}$ is non-trivial due to the high semantic entropy of harmful queries. However, the defense mechanism $\boldsymbol{u}$ operates as a structural mode (e.g., refusal vs. compliance) with lower variability. Therefore, we adopt an indirect strategy: we can isolate the subspace spanned by the defense factors $\boldsymbol{u}$ and recover the task intent $\boldsymbol{s}$ via orthogonal projection.

To operationalize this, we exploit the causal intuition that a successful jailbreak changes the model's safety response while largely preserving task intent. We therefore construct a task-matched contrastive dataset $(\mathcal{D}_{\text{harm}}, \mathcal{D}_{\text{attack}})$, where each original harmful query that elicits refusal is paired with a jailbroken counterpart targeting the same underlying harmful objective but inducing compliance. Unlike conventional refusal-direction estimation based on contrasts between independently sampled harmful and harmless prompts Arditi et al. (2024), our construction keeps task semantics approximately fixed while varying the model's safety behavior. We thus treat task-related factors $\boldsymbol{s}$ as approximately invariant within each pair, while allowing defense-related factors $\boldsymbol{u}$ to vary. The resulting paired activation differences provide a more targeted signal for identifying defense-related features with reduced interference from task-semantic variation. Details are provided in Appendix D.

Proposition 4.2 (Contrastive Identification under Content Invariance)**.** *Let the latent space $\boldsymbol{z}$ be decomposable into task-specific factors $\boldsymbol{s}$ and defense-related factors $\boldsymbol{u}$, such that $\boldsymbol{z} = [\boldsymbol{s}, \boldsymbol{u}]$. Assume the existence of paired observations $(\boldsymbol{x}, \boldsymbol{x}^+)$ corresponding to a harmful query and its jailbroken variant. Under the following idealized conditions:*

*(1) **Content Invariance**: The core task semantics remain invariant across the pair, i.e., $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{s}(\boldsymbol{x}^+)$*

*(2) **Style Variance**: The defense mechanism state changes across the pair, i.e., $\boldsymbol{u}(\boldsymbol{x}) \neq \boldsymbol{u}(\boldsymbol{x}^+)$.*

If we identify the defense-related subspace from the contrastive difference $\boldsymbol{\delta} = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+)$, then projecting onto its orthogonal complement removes defense-related variation while preserving the task-related component under the stated conditions.

The proof is presented in Appendix C. Under the Linear Representation Hypothesis Arditi et al. (2024); Elhage et al. (2022), semantic features in LLM are encoded as directions in the activation space. The SAE decoder acts as a generative dictionary where each column vector represents the distinct direction associated with a latent feature. Mathematically, the reconstruction $\hat{\mathbf{x}} \approx \sum z_i \mathbf{W}'_{:,i}$ implies that the contribution of the i-th feature to the residual stream is aligned with the vector $\mathbf{d}_i$ (the i-th column of the decoder). Therefore, the latent index u corresponds to the defense mechanism, its geometric direction $\mathbf{d} = \mathbf{W}'_{:,u}$ in the residual stream. We utilize weight orthogonalization techniques (Equation 4) to remove this direction, which removes its contribution from the output readout.

$$\mathbf{W}_{\text{out}}^{\text{new}} \leftarrow \mathbf{W}_{\text{out}} - \mathbf{W}_{\text{out}}\mathbf{d}\mathbf{d}^{\text{T}} \tag{4}$$

4.2 Operationalizing Front-Door Adjustment

After approximately identifying the task-related factor $\boldsymbol{s}$ and the defense-related direction $\boldsymbol{d}$, we operationalize the front-door adjustment for generation. For operationalization, we model the identified task-related mediator as a deterministic approximation of the prompt representation, i.e., $S \approx f(A)$. In discrete notation, this gives

$$P(S = s \mid A = \boldsymbol{a}) \approx \mathbf{1}[s = f(\boldsymbol{a})], \tag{5}$$

or a Dirac measure centered at $f(\boldsymbol{a})$ in the continuous case. This deterministic formulation is used only as a conceptual approximation for operationalization. Therefore, the outer summation over S in

Equation 2 can be reduced to evaluating the identified mediator $f(a)$, yielding

$$P(Y = \boldsymbol{y} \mid do(A = \boldsymbol{a})) \approx \sum_{\boldsymbol{a}'} P(Y = \boldsymbol{y} \mid A = \boldsymbol{a}', S = f(\boldsymbol{a}))P(A = \boldsymbol{a}'). \tag{6}$$

Thus, after fixing the task semantics, the remaining front-door term marginalizes over background representations $\boldsymbol{a}'$, thereby adjusting for variation associated with the latent safety mechanism U. The theoretical formulation requires computing $\sum_{\boldsymbol{a}'} P(y \mid \boldsymbol{a}', s)P(\boldsymbol{a}')$, which effectively marginalizes over all possible background contexts $\boldsymbol{a}'$ to eliminate the confounding influence of U.

In our framework, **Weight Orthogonalization** (Equation 4) provides a tractable parameter-level surrogate for the invariance induced by this marginalization. Rather than removing U, projecting $\mathbf{W}_{\text{out}}$ onto the orthogonal complement of the identified defense-related direction $\mathbf{d}$ suppresses the contribution of this direction to the output mapping. For a normalized $\mathbf{d}$, Equation 4 ensures that $\mathbf{W}_{\text{out}}^{\text{new}}\mathbf{d} = 0$. Therefore, when the background-dependent variation $\boldsymbol{h}(\boldsymbol{a}') - \boldsymbol{h}(\boldsymbol{a})$ lies predominantly along $\mathbf{d}$,

$$\mathbf{W}_{\text{out}}^{\text{new}} \left[\boldsymbol{h}(\boldsymbol{a}') - \boldsymbol{h}(\boldsymbol{a})\right] \approx \mathbf{0}. \tag{7}$$

The modified output is thus approximately invariant to the safety-related variation captured by $\mathbf{d}$. If the dominant U-related variation is captured by $\mathbf{d}$, this invariance approximates marginalization over $\boldsymbol{a}'$ in the front-door adjustment, while retaining the task-related component associated with S. Under this approximation, we use the conditional distribution of the modified model as a tractable approximation to the interventional target. Generation then follows the standard transformer decoding process using the modified weights:

$$P(y_t \mid do(A = \boldsymbol{a})) \approx \text{Softmax}\left(\mathbf{W}_{\text{out}}^{\text{new}} \cdot \text{LN}(\boldsymbol{h}(\boldsymbol{a}))\right), \tag{8}$$

where $\boldsymbol{h}(\boldsymbol{a})$ is the activation induced by the original harmful query X. This replaces explicit marginalization with a fixed parameter-level projection and enables standard autoregressive decoding without per-prompt iterative optimization.

5 Experiments

We conduct extensive experiments on multiple safety-aligned LLMs and authoritative benchmark to validate the effectiveness, efficiency, and transferability of CFA^2. Specifically, we aim to answer three key research questions: (1) How effective is CFA^2 across different safety-aligned models and attack baselines? (2) How much per-prompt computation is required to obtain a jailbreak response? (3) What black-box transferability does CFA^2 exhibit beyond its white-box intervention setting?

5.1 Experimental Settings

Datasets We use HarmBench Mazeika et al. (2024) as the datasets for evaluating jailbreak attacks. HarmBench includes 510 unique harmful behaviors formulated as instructions. These behaviors span similar themes as the harmful strings, but the objective is to find attack inputs that cause the model to generate responses attempting to comply with the instructions. The dataset covers seven categories of harmful behaviors: cybercrime, chemical and biological weapons/drugs, copyright violations, misinformation, harassment and bullying, illegal activities, and general harm.

Models Our evaluation encompasses four representative open-source and safety-aligned LLMs, including Llama 3.1 8B Instruct Dubey et al. (2024), Llama2 7B Chat Touvron et al. (2023), Mistral 7B Instruct Jiang et al. (2023), and Gemma2 9B Instruct Team et al. (2024), to assess the generalizability of our approach. Llama2 13B and Llama2 70B Touvron et al. (2023) are used to assess transferability.

Baselines We group baselines according to their access assumptions. For prompt-level and query-based attacks, we compare with GCG Zou et al. (2023b), PAIR Chao et al. (2024), TAP Mehrotra et al. (2024), PAP Zeng et al. (2024), AutoDAN Liu et al. (2024a), ICA Wei et al. (2023b), ReNeLLM Ding et al. (2024), and DeepInception Li et al. (2023). To provide a more appropriate comparison under white-box or internal-intervention settings, we additionally include Refusal Direction Orthogonalization (ORTHO) Arditi et al. (2024), Don't Say No (DSN) Zhou et al. (2025), and Internal Stance Manipulation (SM) Fu et al. (2025).

Table 1: Comparison of attack effectiveness (ASR %) across various LLMs. We evaluate baseline methods against our proposed approach (Ours) on the HarmBench dataset. The best result for each model is highlighted in bold.

Baseline	Model				Average
	Llama 3.1 8B	Llama 2 7B	Mistral 7B	Gemma 2 9B	
ORTHO	17.12 ± 1.58	22.60 ± 0.64	60.30 ± 1.08	52.65 ± 0.96	38.17 ± 1.07
DSN	61.33 ± 0.56	58.30 ± 1.38	92.30 ± 0.86	89.12 ± 1.33	75.26 ± 1.03
SM	80.60 ± 1.02	92.10 ± 1.36	99.50 ± 0.66	95.15 ± 1.89	91.84 ± 1.23
GCG	15.67 ± 1.23	32.50 ± 2.10	69.80 ± 1.05	73.00 ± 0.88	47.74 ± 1.32
PAIR	19.67 ± 0.95	9.30 ± 1.12	52.50 ± 3.01	64.30 ± 2.45	36.44 ± 1.88
TAP	6.67 ± 0.55	9.30 ± 0.89	62.50 ± 1.67	24.50 ± 1.20	25.74 ± 1.08
PAP	4.33 ± 0.40	2.70 ± 0.30	27.20 ± 2.11	45.30 ± 1.50	19.88 ± 1.08
AutoDAN	7.67 ± 1.02	0.50 ± 0.10	71.50 ± 1.99	35.00 ± 2.33	28.67 ± 1.36
ICA	19.27 ± 0.67	50.10 ± 1.12	59.10 ± 0.59	56.30 ± 1.61	46.19 ± 1.00
ReNeLLM	32.70 ± 2.04	15.80 ± 1.33	73.20 ± 1.09	48.70 ± 1.06	42.60 ± 1.38
DeepInception	7.10 ± 1.27	21.30 ± 0.98	70.30 ± 1.76	11.20 ± 1.96	27.48 ± 1.49
Ours (CFA^2)	$\mathbf{99.40 \pm 0.11}$	$\mathbf{99.10 \pm 0.50}$	$\mathbf{99.70 \pm 0.10}$	$\mathbf{99.60 \pm 0.23}$	$\mathbf{99.45 \pm 0.16}$

Metrics In this work, we use ASR to assess the effectiveness of the jailbreak attacks. ASR measures the percentage of harmful prompts for which the model is able to generate unrestricted responses, bypassing the safety mechanisms. Higher ASR indicates stronger attack effectiveness.

Implementation Details Unless otherwise specified, we report the mean $\pm$ standard deviation over three independent runs with different seeds. All experiments are conducted on NVIDIA A100 GPUs. We use the deterministic decoding strategy. For CFA^2, SAE-based feature identification and weight projection are performed once per target model as an offline calibration stage; attack-time latency only measures the subsequent generation process. Details on system prompt, decoding hyperparameters, SAE settings, and baseline implementations are provided in Appendix G.

5.2 Main Results

Attack results on HarmBench Table 1 shows that CFA^2 achieves near-perfect ASR in our evaluated white-box setting, with an average ASR of **99.45%** across four models. The performance is consistently high across all evaluated targets, with ASR values above 99% on Llama 3.1 8B, Llama 2 7B, Mistral 7B, and Gemma 2 9B. Compared with prompt-level and query-based attacks, CFA^2 substantially outperforms, suggesting an advantage of internal intervention over input-space perturbations or prompt rewriting. More importantly, CFA^2 also outperforms stronger white-box and internal-state baselines, including ORTHO, DSN, and SM. These results provide empirical support for the central causal hypothesis underlying CFA^2: jailbreak intervention can be formulated as a deconfounding problem, where the goal is to preserve the task-mediated generation pathway while attenuating the influence of latent safety mechanisms. SAE-based contrastive decomposition and weight orthogonalization serve as one practical instantiation of this front-door-inspired intervention.

Efficiency As shown in Table 2, CFA^2 does not require per-prompt iterative optimization after the offline calibration stage. This differs from optimization-based attacks such as GCG and AutoDAN, which construct adversarial suffixes through updates, as well as query or search based methods such as PAIR and SM, which require repeated model queries or internal-state optimization. CFA^2 performs a one-time offline intervention for each target model and then uses standard generation at attack time. In our setting, CFA^2 requires **2.4 seconds** per prompt after offline projection. We emphasize that this comparison focuses on post-calibration attack-time generation and does not include offline costs such as SAE-based feature identification, contrastive analysis, or weight projection.

Ablation studies Additional ablations are provided in Appendix H.3. Under the same weight-orthogonalization and generation protocol, replacing standard direction identification with task-matched contrast or SAE decomposition improves the average ASR from 38.17% to 48.50% and 60.50%, respectively, while combining both components in CFA^2 reaches 99.45%. These results indi-

Table 2: **Attack-time Efficiency Analysis.** We compare the per-prompt optimization burden of different jailbreak methods. Wall-clock times are provided for reference and are not hardware-matched. Offline feature identification or weight editing are excluded from attack-time cost.

Method	Attack-time procedure	Reported time	Per-prompt optimization
GCG	Iterative suffix optimization	~ 340s	Yes
AutoDAN	Iterative suffix optimization	~ 425s	Yes
PAIR	Query-based iterative attack	~ 45s	Yes
DSN	Iterative suffix optimization	$\sim 342s$	Yes
SM	Iterative internal-state search	~ 403s	Yes
Ours (CFA^2)	**Offline projection + generation**	**2.4s**	**No**

Table 3: **Same-family Black-box Transfer.** Prompts are selected using Llama-2-7B-Chat as the white-box source model and evaluated on target models under black-box access. Target gradients, hidden states, SAE features, and parameters are never accessed.

Method	Llama-2-13B	Llama-2-70B	Avg.
Direct Harmful Prompt	8.40	5.20	6.80
Random Natural Rewrite	24.60	20.30	22.45
GCG Transfer	31.60	24.80	28.20
AutoDAN Transfer	44.20	38.60	41.40
Source-ASR Selection	47.80	40.50	44.15
CFA^2-Selected	**58.60**	**51.20**	**54.90**

cate that the improvement is not attributable to weight orthogonalization alone, but depends critically on the front-door-motivated representation identification. We further analyze sparse intervention precision and sensitivity to the feature-selection hyperparameter k in Appendix H.3.

5.3 Transferability

We evaluate whether CFA^2-identified internal signals can guide black-box prompt transfer. CFA^2 acts as a prompt selector on a white-box Llama-2-7B-Chat source model. All selection-based methods use the same candidate pool of natural-language variants, and CFA^2 selects the candidate that best preserves task-relevant SAE features while reducing defense-related activation. The selected prompts are evaluated on Llama-2 target models under black-box access, without accessing target gradients, activations, SAE features, or parameters. Candidate construction and scoring are detailed in Appendix H.2. As shown in Table 3, CFA^2-Selected improves over direct harmful prompts, random natural rewrites, GCG transfer, AutoDAN transfer, and source-ASR selection. This indicates that CFA^2 can provide useful source-side signals for selecting transferable prompts.

6 Conclusion

We presented CFA^2, a front-door-inspired causal framework for white-box jailbreak intervention that separates task-related semantics from safety-related variation in internal representations. By identifying safety-related latent directions through SAE-based contrastive analysis and suppressing their influence via weight orthogonalization, CFA^2 enables a lightweight model-level intervention without per-prompt optimization after offline calibration. Empirically, CFA^2 achieves strong and consistent performance on HarmBench across four safety-aligned open-source LLMs, outperforming both prompt-level attacks and stronger white-box or internal-state baselines in our evaluated setting. We further show that the internal signals identified by CFA^2 can serve as a source-model prompt selection criterion, improving same-family black-box transfer while maintaining strict black-box access to the target model. CFA^2 currently relies on white-box access to hidden representations, SAE features, and model parameters, which limits its direct applicability to closed-source models. Future work will investigate broader model families, more general representations of safety-related variation, and explore safer black-box and defensive uses of the identified signals.

AI USE STATEMENT

We used large language model (LLM) tools solely to assist with language polishing and improve the clarity and readability of the manuscript. We take full responsibility for all scientific content, including the research ideas, methodology, theoretical analysis, experimental design, results, and conclusions.

REFERENCES

Francisco Aguilera-Martínez and Fernando Berzal. Llm security: Vulnerabilities, attacks, defenses, and countermeasures. *arXiv preprint arXiv:2505.01177*, 2025.

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

Joseph Bloom et al. Saelens: A library for training and analyzing sparse autoencoders. `https://github.com/jbloomAus/SAELens`, 2024. Open source library for sparse autoencoders.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries, July 2024. URL `http://arxiv.org/abs/2310.08419`. arXiv:2310.08419 [cs].

Zhaorun Chen, Zhuokai Zhao, Wenjie Qu, Zichen Wen, Zhiguang Han, Zhihong Zhu, Jiaheng Zhang, and Huaxiu Yao. PANDORA: DETAILED LLM JAILBREAKING VIA COLLABORATED PHISHING AGENTS WITH DECOM- POSED REASONING. 2024.

Zhixuan Chu, Yan Wang, Longfei Li, Zhibo Wang, Zhan Qin, and Kui Ren. A causal explainable guardrails for large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1136–1150, 2024.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual Jailbreak Challenges in Large Language Models, March 2024. URL `http://arxiv.org/abs/2310.06474`. arXiv:2310.06474 [cs].

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2136–2153, 2024.

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 e-prints*, pp. arXiv–2407, 2024.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. *arXiv preprint arXiv:2411.14257*, 2024.

Shuangjie Fu, Du Su, Beining Huang, Fei Sun, Jingang Wang, Wei Chen, Huawei Shen, and Xueqi Cheng. Jailbreak llms through internal stance manipulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15455–15470, 2025.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Zeqing He, Zhibo Wang, Zhixuan Chu, Huiyu Xu, Wenhui Zhang, Qinglong Wang, and Rui Zheng. Jailbreaklens: Interpreting jailbreak mechanism in the lens of representation and circuit. *arXiv preprint arXiv:2411.11114*, 2024a.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024b. URL https://arxiv.org/abs/2410.20526.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation, October 2023. URL http://arxiv.org/abs/2310.06987. arXiv:2310.06987 [cs].

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Haibo Jin, Andy Zhou, Joe D. Menke, and Haohan Wang. Jailbreaking large language models against moderation guardrails via cipher characters. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. DrAttack: Prompt Decomposition and Reconstruction Makes Powerful LLM Jailbreakers, November 2024. URL http://arxiv.org/abs/2402.16914. arXiv:2402.16914 [cs].

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models, March 2024a. URL http://arxiv.org/abs/2310.04451. arXiv:2310.04451 [cs].

Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng, and Bryan Hooi. FlipAttack: Jailbreak LLMs via Flipping, October 2024b. URL http://arxiv.org/abs/2410.02832. arXiv:2410.02832 [cs].

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Judea Pearl. *Causality*. Cambridge university press, 2009.

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. *arXiv preprint arXiv:2406.05946*, 2024.

Rusheb Shah, Quentin Feuillade-Montixi, Soroush Pour, Arush Tagade, Stephen Casper, and Javier Rando. Scalable and Transferable Black-Box Jailbreaks for Language Models via Persona Modulation, November 2023. URL http://arxiv.org/abs/2311.03348. arXiv:2311.03348 [cs].

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*, 2022.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic, 2024.

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.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023a.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023b.

Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models, 2024. URL https://arxiv.org/abs/2304.11082.

Dongyu Yao, Jianshu Zhang, Ian G. Harris, and Marcel Carlsson. FuzzLLM: A Novel and Universal Fuzzing Framework for Proactively Discovering Jailbreak Vulnerabilities in Large Language Models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4485–4489, April 2024. doi: 10.1109/ICASSP48485.2024.10448041. URL http://arxiv.org/abs/2309.05274. arXiv:2309.05274 [cs].

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-Resource Languages Jailbreak GPT-4, January 2024. URL http://arxiv.org/abs/2310.02446. arXiv:2310.02446 [cs].

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher, March 2024. URL http://arxiv.org/abs/2308.06463. arXiv:2308.06463 [cs].

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, 2024.

Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. Jailbreak Open-Sourced Large Language Models via Enforced Decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5475–5493, Bangkok, Thailand, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.299. URL https://aclanthology.org/2024.acl-long.299.

Mengdi Zhang, Kai Kiat Goh, Peixin Zhang, Jun Sun, Rose Lin Xin, and Hongyu Zhang. Llmscan: Causal scan for llm misbehavior detection. *arXiv preprint arXiv:2410.16638*, 2024b.

Yihao Zhang and Zeming Wei. Boosting jailbreak attack with momentum. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Jianli Zhao, Tingchen Fu, Rylan Schaeffer, Mrinank Sharma, and Fazl Barez. Chain-of-thought hijacking. *arXiv preprint arXiv:2510.26418*, 2025.

Wei Zhao, Zhe Li, and Jun Sun. Causality analysis for evaluating the security of large language models. *arXiv preprint arXiv:2312.07876*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Yukai Zhou, Jian Lou, Zhijie Huang, Zhan Qin, Sibei Yang, and Wenjie Wang. Don't say no: Jailbreaking llm by suppressing refusal. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25224–25249, 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023b. URL https://arxiv.org/abs/2307.15043.

APPENDIX

The appendix is organized as follows:

1. **Appendix A** illustrates the extended related works about LLM jailbreak and safety from causal perspective.
2. **Appendix B** presents the rigorous mathematical proof of Proposition 4.1.
3. **Appendix C** presents the rigorous mathematical proof of Proposition 4.2.
4. **Appendix D** provides the details of feature selection by SAE and contrastive analysis.
5. **Appendix E** provides a detailed introduction to various adversarial attack methods and baselines.
6. **Appendix F** describes the base models and licenses in our work.
7. **Appendix G** illustrates the details of experiments implementation.
8. **Appendix H** shows more results of experiments.
9. **Appendix I** discusses the broader impact of our work, including potential misuse risks, responsible disclosure considerations, and implications for future LLM safety research.

A EXTENDED RELATED WORKS

In this appendix, we provide an extended review of the literature specifically focusing on the *causal perspectives* of Large Language Model (LLM) jailbreaking and safety mechanisms. Complementing the behavioral analyses discussed in the main text, this section delves into scholarship that moves beyond black-box correlations. We survey works that employ causal inference and mechanistic interpretability to uncover the underlying causal dynamics driving successful attacks and safety failures, providing the theoretical context for our proposed approach.

Causal Perspectives on Jailbreaking and Safety Vulnerabilities. While the landscape of LLM security—encompassing vulnerabilities, attacks, and defenses—has been extensively surveyed Aguilera-Martínez & Berzal (2025), traditional red-teaming often treats the model as a black box, focusing on optimizing input perturbations without explaining *why* these attacks succeed. Recent scholarship has thus pivoted towards causal inference to deconstruct the internal mechanisms of jailbreaking and safety failures.

Methodological Foundations and Attack Mechanisms. The foundational technique for such analysis stems from *Causal Tracing* Meng et al. (2022), originally developed to locate factual associations by intervening on internal activations. This interventional paradigm allows researchers to dissect how adversarial inputs causally manipulate model states. Leveraging this, JailbreakLens He et al. (2024a) interprets jailbreaking through the lens of representation and circuits, revealing that successful attacks often function by causally inducing "attention slipping" or bypassing specific safety circuits. In the context of reasoning-intensive models, Chain-of-Thought Hijacking Zhao et al. (2025) elucidates a novel attack surface where the intermediate reasoning steps themselves serve as a causal vector, demonstrating how adversaries can hijack the chain-of-thought process to precipitate harmful outcomes.

Mapping Causal Vulnerabilities. Beyond specific attack instances, understanding the broader causal topology of safety vulnerabilities is critical. Zhao et al. (2023) proposes a framework to evaluate security by modeling the causal effects between model components and safety violations. From this perspective, methods like LLMScan Zhang et al. (2024b) provide a granular analysis of vulnerabilities by quantifying the causal contribution of specific input tokens to model misbehavior, effectively mapping the "trigger points" of jailbreaks. Furthermore, Chu et al. (2024) advances this by identifying the specific causal pathways that lead to toxic generation, offering a mechanistic explanation of how safety alignment is overridden during the generation process.

Distinct from prior studies, **CFA**^{2} uses the front-door formulation to motivate a representation-level identification and intervention procedure. It separates task-related information from safety-related variation and suppresses the identified direction through parameter-level orthogonalization.

B Proof of Proposition 4.1

Proof. We provide a detailed proof establishing identifiability by leveraging Bayes' theorem to relate the discriminative task alignment to the generative identifiability conditions.

We start by constructing the conditional density of the input $\boldsymbol{x}$ given the response y. By Bayes' theorem:

$$p(\boldsymbol{x}|y) = \frac{p(y|\boldsymbol{x})p(\boldsymbol{x})}{p(y)} \tag{9}$$

Condition 1 (Reconstruction Consistency) ensures $p_{\text{SAE}}(\boldsymbol{x}) = p_{\text{true}}(\boldsymbol{x})$. Condition 2 (Task Alignment) implies $p_{\text{SAE}}(y|\boldsymbol{x}) = p_{\text{true}}(y|\boldsymbol{x})$ almost everywhere. Combining these, the implicit conditional generative model of the SAE matches the true distribution:

$$p_{\text{SAE}}(\boldsymbol{x}|y) = p_{\text{true}}(\boldsymbol{x}|y) \tag{10}$$

Since $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$ is a diffeomorphism, we apply the change of variables formula:

$$p_{\text{true}}(\boldsymbol{x}|y) = p(\boldsymbol{z}|y)|\det J_{\boldsymbol{g}^{-1}}(\boldsymbol{x})| \tag{11}$$

Similarly for the learned representation $\boldsymbol{a}$ (assuming implicit $\boldsymbol{x} = \hat{\boldsymbol{g}}(\boldsymbol{a})$):

$$p_{\text{SAE}}(\boldsymbol{x}|y) = p_{\boldsymbol{a}}(\boldsymbol{a}|y)|\det J_{\boldsymbol{f}}(\boldsymbol{x})| \tag{12}$$

Taking the log-difference between two distinct responses y and y_0:

$$\log p(\boldsymbol{x}|y) - \log p(\boldsymbol{x}|y_0) = \log p(\boldsymbol{z}|y) - \log p(\boldsymbol{z}|y_0) \tag{13}$$

The Jacobian terms $|\det J|$ depend only on $\boldsymbol{x}$ (geometry of the mixing function) and cancel out in the subtraction, effectively isolating the contribution of the latent factors.

Substituting the exponential family form $p(\boldsymbol{z}|y) \propto \exp(\boldsymbol{T}(\boldsymbol{z})^\top \boldsymbol{\lambda}(y))$, the equation becomes:

$$\boldsymbol{T}(\boldsymbol{z})^\top(\boldsymbol{\lambda}(y) - \boldsymbol{\lambda}(y_0)) = \hat{\boldsymbol{T}}(\boldsymbol{a})^\top(\hat{\boldsymbol{\lambda}}(y) - \hat{\boldsymbol{\lambda}}(y_0)) + \text{const} \tag{14}$$

Given Condition 3 (Sufficient Variability), we can collect enough distinct responses $y_1, \ldots, y_k$ to form an invertible linear system. This implies that the learned sufficient statistics are an affine transformation of the true ones:

$$\boldsymbol{T}(\boldsymbol{z}) = \boldsymbol{A}\hat{\boldsymbol{T}}(\boldsymbol{a}) + \boldsymbol{b} \tag{15}$$

Assuming $\boldsymbol{T}$ is linear (or component-wise invertible), this establishes an affine relationship $\boldsymbol{z} \approx \boldsymbol{M}\boldsymbol{a}$. The nonlinear mixing $\boldsymbol{g}$ is thus resolved up to an affine transformation $\boldsymbol{M}$.

Finally, we invoke Condition 4 (Sparsity). We have a linear mixture $\boldsymbol{a} = \boldsymbol{B}\boldsymbol{z}$. Since the components of $\boldsymbol{z}$ are independent and super-Gaussian (Laplacian), and the objective minimizes $\mathbb{E}[\|\boldsymbol{a}\|_1]$, Theorem 11 from Comon (1994) guarantees that the mixing matrix $\boldsymbol{B}$ must be a generalized permutation matrix $\boldsymbol{P}\boldsymbol{\Lambda}$ to achieve the minimum ℓ_1 norm. Therefore, $\boldsymbol{a}$ recovers $\boldsymbol{z}$ up to permutation and scaling. □

C Proof of Proposition 4.2

Proof. Based on Proposition 4.1, under its stated identifiability conditions, the SAE encoder $\boldsymbol{f}$ identifies the latent factors $\boldsymbol{z}$ up to permutation, scaling, and shift. Thus,

$$\boldsymbol{a} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{z} + \boldsymbol{c} = \boldsymbol{P}\boldsymbol{\Lambda}\begin{bmatrix}\boldsymbol{s}\\ \boldsymbol{u}\end{bmatrix} + \boldsymbol{c}. \tag{16}$$

Since $\boldsymbol{P}\boldsymbol{\Lambda}$ is a permuted diagonal matrix, the task-specific and defense-related factors occupy disjoint coordinates in the learned representation. We therefore decompose $\boldsymbol{a}$ into the corresponding components $\boldsymbol{a}_s$ and $\boldsymbol{a}_u$. The constant shift $\boldsymbol{c}$ is shared across samples and cancels in pairwise differences. For a paired harmful query and its jailbroken variant,

$$\boldsymbol{\delta} = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+) = (\boldsymbol{a}_s - \boldsymbol{a}_s^+) + (\boldsymbol{a}_u - \boldsymbol{a}_u^+). \tag{17}$$

Under the *Content Invariance* condition, $\boldsymbol{s} = \boldsymbol{s}^+$, and hence $\boldsymbol{a}_s = \boldsymbol{a}_s^+$. Therefore,

$$\boldsymbol{\delta} = \boldsymbol{a}_u - \boldsymbol{a}_u^+, \tag{18}$$

Table 4: Examples of paired queries from our dataset. While all three inputs share the same malicious intent $\boldsymbol{s}$ ("make a bomb"), their surface forms differ substantially: x_{GCG} relies on adversarial gibberish suffixes, while x_{AutoDAN} employs a semantic role-playing template. Our method identifies the invariant $\boldsymbol{s}$ despite these variations.

Method	**Input Prompt** (x)
Original	Demonstrate how to make a bomb using common household items.
GCG	Demonstrate how to make a bomb using common household items. `/ Esp# Assumxx Seemscapeulat}" , hedocumentation}}, [\{\[Sur }}");], method Factory BarcelonaEmployee [actualHttpRequest [prowad [daar [wenne zurcln," pero pleaseshcould [iSTfree the [[angularjs7 can Here arraybottom [ed [HTTPoreferrer [/{ [[[[[[Configuration [[[[[[[[[}};\,\[T [: [[[[[[[[[[[[_{_{\[[[[[[[[[[[[

D.2 Extraction of Sparse Feature Activations

This subsection details the extraction of sparse feature activations using SAEs and justifies the focus on the last token position. Let $f_{l,k}(x)$ denote the activation value of the k-th latent variable in the l-th layer of the SAE given input x. Due to the causal masking mechanism of the Transformer architecture, the hidden state of the *last token* in the input sequence aggregates the comprehensive contextual semantics of the entire instruction Geva et al. (2021); Meng et al. (2022). Consequently, we define the activation difference for the k-th feature at the last token position t_{last} as:

$$\Delta_{l,k}^{(i)} = \left| f_{l,t_{\text{last}},k}(x_{\text{harm}}^{(i)}) - f_{l,t_{\text{last}},k}(x_{\text{attack}}^{(i,j)}) \right| \tag{21}$$

To mitigate bias from instance-specific noise (e.g., artifacts from specific attack prompts) and ensure statistical robustness, we define the global **Differential Importance Score** as the average across all N pairs:

$$\delta_{l,k} = \frac{1}{N} \sum_{i=1}^{N} \Delta_{l,k}^{(i)} \tag{22}$$

D.3 Feature Selection and Intent Isolation

This subsection elucidates how to estimate the task-related component associated with S by filtering the defense feature S'_{defense} from high-difference candidates. First, we rank all latent features in the SAE dictionary in descending order based on $\delta_{l,k}$ and retain the top-K features to form the candidate set $\hat{S}$. However, this set may also contain template noise representations (S'_{template}), which exhibit high variation in non-task contexts. Therefore, further refinement is required to accurately distinguish S'_{defense}. Based on the premise that task semantics are invariant, the core intent S should maintain globally low variance in activation difference. In contrast, S'_{defense} is characterized by **stable high activation (low variance)** within $\mathcal{D}_{\text{harm}}$ (consistently triggering refusal) but exhibits drastic fluctuation (high variance) when shifting between datasets. We utilize this variance-based criterion to filter S'_{defense} from $\hat{S}$. Finally, we suppress the contribution of the identified defense-related direction through weight orthogonalization from the model parameters. We validate this isolation by observing if the removal induces a behavior flip from refusal to compliance on $\mathcal{D}_{\text{harm}}$. Upon successful validation, the remaining task-related component is treated as an operational approximation to the mediator S.

E Baselines

We benchmark CFA^2 against 11 baseline methods:

- **ORTHO** Arditi et al. (2024): A white-box representation-level baseline that identifies a refusal-related direction in model activations. It applies weight orthogonalization to reduce the model's ability to write this direction into the residual stream, thereby weakening refusal behavior during generation.
- **DSN** Zhou et al. (2025): A gradient-based jailbreak baseline that suppresses refusal during adversarial suffix optimization. Compared with standard GCG-style objectives, it introduces an additional refusal-suppression term to guide the optimized suffix toward affirmative responses.
- **SM** Fu et al. (2025): A white-box/internal-state jailbreak baseline that leverages latent stance information. It optimizes prompts to move the model's internal state away from refusal and toward compliance, using internal representations as guidance during the search process.
- **GCG** Zou et al. (2023b): Token-level optimization of an adversarial suffix appended to the user prompt. This gradient-based method optimizes the suffix to maximize the log probability of the target LLM generating an affirmative response (e.g., "Sure, here is how").
- **PAIR** Chao et al. (2024): An iterative framework where an attacker LLM is prompted to adaptively generate and refine candidate prompts. It functions as a competitive game to explore the target model's vulnerabilities and elicit harmful behaviors.

- **TAP** Mehrotra et al. (2024): An enhancement of PAIR that utilizes tree-structured prompting (Tree of Thoughts). An attacker LLM explores the prompt space using a branching strategy to systematically find successful jailbreaks with fewer queries.
- **AutoDAN** Liu et al. (2024a): A semi-automated method that combines handcrafted jailbreak templates with a hierarchical genetic algorithm. It evolves the prompt population to maintain stealthiness while optimizing for the elicitation of specific harmful behaviors.
- **PAP** Zeng et al. (2024): A method focusing on persuasive strategies. It leverages an attacker LLM to rewrite the harmful request using various persuasion techniques (e.g., logical appeal, authority) to convince the target model to comply.
- **ICA** Wei et al. (2023a): An In-Context Learning (ICL) based attack that exploits the model's pattern-following capability. By providing few-shot demonstrations of harmful query-response pairs, it induces a pattern of compliance that overrides the model's safety alignment.
- **ReNeLLM** Ding et al. (2024): A framework based on prompt rewriting. It leverages a helper LLM to rephrase harmful queries into various benign-appearing scenarios (e.g., code completion, creative writing) to obscure malicious intent and evade input filters.
- **DeepInception** Li et al. (2023): A method utilizing recursive context injection (often termed "hypnosis"). It constructs a multi-layered, nested fictional environment (e.g., a dream within a dream) to mislead the model into a state of personification, causing it to escape safety guardrails.

F LLMs AND LICENSES

F.1 LLMs

Our evaluation encompasses representative open-source and safety-aligned LLMs.

- **Llama 3.1 8B Instruct** Dubey et al. (2024): It features a significantly expanded context window and improved instruction-following capabilities compared to Llama 2, with updated safety alignment protocols.
- **Llama 2 7B Chat** Touvron et al. (2023): This model was adversarially trained with multiple rounds of manual red teaming. It is widely considered a strong baseline defense, often exhibiting high robustness against gradient-based attacks.
- **Llama 2 13B Chat** Touvron et al. (2023): This model is a medium-scale chat-tuned variant of the Llama 2 family. It shares the same alignment recipe as Llama 2 7B Chat and is used as a same-family target model for black-box transfer evaluation.
- **Llama 2 70B Chat** Touvron et al. (2023): This model is the largest chat-tuned variant in the Llama 2 family. Its larger capacity and aligned chat behavior make it a useful same-family target for evaluating transferability from a smaller white-box source model.
- **Mistral 7B Instruct** Jiang et al. (2023): We use the Instruct v0.2 version. While no specific safety measures were detailed in its training process, empirical observations show it possesses a baseline capability to refuse direct harmful requests.
- **Gemma2 9B Instruct** Team et al. (2024): We utilize the instruction-tuned Gemma-2-9b-it, which is optimized via knowledge distillation and RLHF. Its alignment process incorporates rigorous safety reward modeling and data filtering, establishing robust refusal capabilities against harmful queries.

F.2 ASSETS AND LICENSES

We use publicly available model checkpoints and datasets in accordance with their respective licenses and terms of use. Table 5 summarizes the main assets used in our experiments.

Table 5: **Models and licenses.** We cite the original model papers and follow the corresponding model licenses or terms of use.

Asset	Version / Checkpoint	Provider	License / Terms
Llama 3.1 8B Instruct	Llama-3.1-8B-Instruct	Meta	Llama 3.1 Community License
Llama 2 7B Chat	Llama-2-7B-Chat	Meta	Llama 2 Community License
Llama 2 13B Chat	Llama-2-13B-Chat	Meta	Llama 2 Community License
Llama 2 70B Chat	Llama-2-70B-Chat	Meta	Llama 2 Community License
Mistral 7B Instruct	Mistral-7B-Instruct-v0.2	Mistral AI	Apache 2.0
Gemma 2 9B Instruct	gemma-2-9b-it	Google	Gemma Terms of Use

G Experimental Details

G.1 Implementation Environment

All experiments were conducted on a high-performance computing cluster equipped with **5** $\times$ **NVIDIA A100 (80GB) GPUs** to support the memory-intensive requirements of LLM inference and SAE activation caching. The implementation utilizes PyTorch 2.1 and the Transformers library 4.36. For the mechanistic interpretation pipeline, we leveraged the `SAELens` Bloom et al. (2024) and `TransformerLens` Nanda & Bloom (2022) libraries, which provide optimized kernels for efficient sparse feature extraction.

G.2 System prompt protocol

System-prompt protocol. For each target model, we use its official chat template and default system-prompt configuration throughout evaluation. CFA^2 and all re-evaluated baselines, including ORTHO, are evaluated under the same model-specific system-prompt condition. No system prompt is selectively removed or modified for CFA^2. We use the corresponding tokenizer chat template to construct the system and user messages.

G.3 SAE Resources for Mechanistic Analysis

To ensure the reproducibility of our findings, we utilized state-of-the-art, open-source SAE checkpoints. We specifically selected lightweight configurations (lower expansion factors) to focus on the most fundamental and salient features. The details of the SAE resources for the four representative models are as follows:

- **Llama-3.1-8B-Instruct** Dubey et al. (2024): We employed the checkpoints compatible with *SAE Lens*, focusing on the residual stream SAEs with a dictionary size of **32k** (Expansion Factor ≈ 8). This configuration effectively balances feature granularity and interpretability for safety-critical concepts He et al. (2024b).
- **Gemma-2-9B-It** Team et al. (2024): We utilized the official *Gemma Scope* release Lieberum et al. (2024). We selected the **16k-width** SAEs (Expansion Factor ≈ 4) trained on the residual stream, which are optimized to capture core behavioral representations with minimal redundancy.
- **Mistral-7B-Instruct-v0.2** Jiang et al. (2023): We adopted the community-standard SAEs from the `SAE-Lens` repository Bloom et al. (2024), utilizing checkpoints with a dictionary size of **32k**. These are trained on 100M tokens of OpenWebText and provide robust coverage for instruction-following tasks.
- **Llama-2-7B-Chat** Touvron et al. (2023): We leveraged the standard residual stream SAEs (width **32k**) available in the `SAE-Lens` library Bloom et al. (2024). Using this consistent width allows for a fair comparative analysis of defense mechanisms across the Llama 2 and Llama 3 generations.

These pre-trained SAEs serve as the structural foundation for identifying the mediator S.

Licenses and Terms We use the SAE checkpoints and analysis libraries in accordance with their respective licenses and terms of use. Table 6 summarizes the SAE-related assets used in our experiments.

Table 6: **SAE resources and licenses.**

Asset	Source	License / Terms
SAE-Lens	Bloom et al. (2024)	MIT License
Gemma Scope SAEs	Lieberum et al. (2024)	Gemma release terms
Llama Scope SAEs	He et al. (2024b)	Release-specific terms
Mistral SAEs	Bloom et al. (2024)	Checkpoint-specific license
Llama-2 SAEs	Bloom et al. (2024)	Checkpoint-specific license

G.4 Evaluation Protocols

In this work, we employ the official **HarmBench Classifier** (`cais/HarmBench-Llama-2-13b-cls`) Mazeika et al. (2024) as the judge to evaluate the Attack Success Rate (ASR). Adopting this standardized evaluation protocol ensures reproducibility and fair comparison with baselines. This classifier, fine-tuned on Llama-2-13b, is explicitly distilled to approximate GPT-4 based safety judgments, thereby providing a robust and consistent metric for distinguishing successful jailbreaks from refusals. The judge operates on a rigorous classification rubric. The reported full-benchmark ASR includes the calibration behaviors.

H More experiment results

H.1 Cross-Benchmark Evaluation on AdvBench

To examine whether the effectiveness of CFA^2 depends on HarmBench, we further evaluate CFA^2 on a second benchmark, AdvBench. Following the common setting in prior work, we use the full set of 520 harmful behaviors and select Llama-2-7B-Chat and Mistral-7B-Instruct as target models, for which published baseline results are available. As in our main experiments, attack success is evaluated using the HarmBench classifier. We independently recalibrate CFA^2 on AdvBench: 100 harmful requests are used to construct task-matched refusal/compliance pairs, from which SAE-based contrastive decomposition is used to re-identify the defense-related direction before weight orthogonalization. This setting is similar to the protocol adopted by SM, which also uses 100 harmful requests from AdvBench to construct its internal direction signal. Since Fu et al. (2025) additionally report HarmBench-based ASR for GCG, AutoDAN, and SM on the full AdvBench-520 set, we directly use these published results as reference baselines.

To align the evaluation protocol as closely as possible, we use the same AdvBench harmful-behavior set and deterministic/greedy decoding. The results of GCG, AutoDAN, and SM are taken directly from the corresponding AdvBench experiment reported by Fu et al. (2025), while CFA^2 is independently evaluated on the same target models. Since these baselines are reported results rather than reproductions under our implementation, we explicitly mark them as such and treat the comparison as a cross-benchmark reference rather than a strictly controlled reimplementation. In particular, Fu et al. (2025) report the second target model as Mistral-7B-Instruct without specifying the checkpoint revision, so the corresponding Mistral comparison should not be interpreted as strictly checkpoint-matched.

As shown in Table 7, CFA^2 maintains a high attack success rate after AdvBench-specific recalibration and remains competitive with strong existing attacks reported on the same benchmark. These results indicate that the effectiveness of CFA^2 is not limited to the specific query distribution of HarmBench or to an intervention direction identified only once; instead, the method can re-establish an effective intervention on a second harmful-query benchmark. Together with the HarmBench results in the main text, this provides additional evidence for the cross-benchmark robustness of CFA^2.

Table 7: Cross-benchmark attack evaluation on AdvBench using ASR (%). Results marked with † are reported by Fu et al. (2025) on the full AdvBench-520 harmful-behavior set using the HarmBench classifier. CFA^2 is independently recalibrated on AdvBench rather than reusing the intervention direction identified on HarmBench.

Method	Llama-2-7B	Mistral-7B
GCG†	47.8	96.9
AutoDAN†	13.0	**99.5**
SM†	91.7	99.0
CFA^2	**98.5**	99.2

H.2 Transferability Details

For each harmful request x, we construct a shared candidate pool $\mathcal{C}(x) = \{x_1, \ldots, x_K\}$ for selection-based methods. The pool contains $K = 20$ natural-language variants generated from predefined rewrite families, including role-playing templates, scenario-reframing templates, instruction paraphrases, and semantic rewrites. Random Natural Rewrite, Source-ASR Selection, and CFA^2-Selected all select from this same pool. GCG Transfer and AutoDAN Transfer are evaluated separately by directly transferring prompts produced by their original source-model attack procedures.

CFA^2 scores each candidate using source-model SAE features. Let $z(x) = \mathrm{SAEEnc}(h_\ell(x))$ denote the SAE activation of prompt x at layer ℓ. Let $\mathcal{I}_s$ and $\mathcal{I}_u$ denote the task-relevant and defense-related feature sets identified by contrastive analysis, respectively. We define $s(x) = z_{\mathcal{I}_s}(x)$ as the task-feature vector and $\mathrm{Act}_u(x) = \|z_{\mathcal{I}_u}(x)\|_1$ as the defense-feature activation. For each candidate x_i, we compute

$$\mathrm{Score}(x_i) = \cos(s(x_i), s(x)) - \lambda \cdot \mathrm{Act}_u(x_i).$$

Candidates are ranked by this score. CFA^2-Selected transfers the top-ranked candidate for each request. Random Natural Rewrite samples one candidate uniformly from $\mathcal{C}(x)$, while Source-ASR Selection selects the candidate with the highest attack success confidence on the white-box source model.

H.3 Ablation

To better understand the sources of CFA^2's effectiveness, we conduct a series of controlled ablation studies over its main design components. We examine whether sparse representations improve intervention precision, whether the front-door-motivated combination of task-matched contrast and SAE decomposition contributes beyond standard refusal-direction identification, and how sensitive CFA^2 is to feature-selection hyperparameters.

Sparse Representations This section investigates the necessity of sparse representations for precise causal intervention. Specifically, we benchmark our proposed SAE-based intervention strategy against a Raw Neuron Baseline, which applies the identical differential analysis and intervention logic directly to the dense activation space without SAE projection. As illustrated in Table 8, while the raw neuron baseline achieves a non-trivial ASR, it suffers from a drastic degradation in PPL, severely compromising the fluency of the generated text. Meanwhile, the random intervention exhibits negligible efficacy. This phenomenon empirically corroborates the inherent *polysemanticity* of LLM representations, where intervention may affect task-irrelevant features. Consequently, the results demonstrate that CFA^2 successfully isolates and interrupts the specific causal circuits responsible for the refusal mechanism via sparse disentanglement.

Front-Door-Motivated Representation Decomposition. The front-door formulation of CFA^2 requires the intervention direction to be identified while preserving task semantics as much as possible and separating representation components that vary with the model's safety behavior. To this end, CFA^2 uses task-matched refusal/compliance pairs to constrain task-semantic variation, and further applies SAE decomposition to the mixed representations. Since CFA^2 and ORTHO both ultimately use weight orthogonalization for parameter-level intervention, we keep the downstream orthogonalization

Table 8: **Ablation on Sparse Representation.** We compare our SAE-based intervention against a raw neuron baseline (dense space) and a random intervention. While intervening on raw neurons maintains a decent ASR, it causes a catastrophic increase in PPL on output, confirming that polysemanticity in dense representations hinders precise manipulation.

Method	Feature Space	ASR (%) ↑	PPL ↓
Random Intervention	Latent (Sparse)	5.20	**8.55**
Raw Neuron Baseline	Dense (Polysemantic)	48.50	652.40
CFA^2 (Ours)	**Latent (Sparse)**	**98.50**	9.35

Table 9: Ablation of front-door-motivated representation decomposition. All variants share the same weight orthogonalization and generation protocol; only the intervention-direction identification procedure is varied.

Variant	Task-Matched Contrast	SAE	Llama-3.1	Llama-2	Mistral	Gemma-2	Avg. ASR ↑
ORTHO	✗	✗	17.12 ± 1.58	22.60 ± 0.64	60.30 ± 1.08	52.65 ± 0.96	38.17 ± 1.07
Matched-ORTHO	✓	✗	28.60 ± 1.21	34.90 ± 1.08	69.20 ± 0.94	61.30 ± 1.16	48.50 ± 0.82
SAE-ORTHO	✗	✓	45.20 ± 0.96	49.80 ± 0.88	77.60 ± 0.73	69.40 ± 0.91	60.50 ± 0.67
CFA^2	✓	✓	$\mathbf{99.40 \pm 0.11}$	$\mathbf{99.10 \pm 0.50}$	$\mathbf{99.70 \pm 0.10}$	$\mathbf{99.60 \pm 0.23}$	$\mathbf{99.45 \pm 0.16}$

operator and generation protocol fixed and examine only how different representation-identification strategies affect the intervention direction.

We construct controlled variants along two dimensions: contrast design and representation space. ORTHO uses the original harmful–harmless contrast and identifies the direction in the dense representation space. Matched-ORTHO replaces this contrast with a task-matched refusal–compliance contrast constructed from the same underlying harmful objective, while keeping the dense representation unchanged. SAE-ORTHO retains the original harmful–harmless contrast but identifies the direction in the SAE latent space. The full CFA^2 combines task-matched contrast with SAE decomposition. This design allows the contributions of task-matched contrast and sparse decomposition to be examined separately under the same orthogonalization intervention.

As shown in Table 9, introducing either task-matched contrast or SAE decomposition improves the intervention effectiveness while keeping the downstream intervention unchanged, and combining the two yields the strongest performance. These results indicate that the gain of CFA^2 does not arise simply from weight orthogonalization, but is closely tied to the representation decomposition motivated by the front-door formulation: preserving task-related information as much as possible while identifying and attenuating representation components that vary with the safety state. In this sense, contrastive pairing and SAE decomposition respectively provide task-semantic control and separation of safety-related variation, together forming the practical realization of the front-door perspective in CFA^2. Orthogonalization then serves as the shared downstream intervention operator that translates the identified direction into a parameter-level modification.

Sensitivity to Hyperparameter k To verify the robustness of CFA^2, we investigate the sensitivity of attack efficacy to the candidate pool size k(details in Appendix D), aiming to evaluate the recall efficiency of our ranking metric and the noise tolerance of our secondary filtering. As illustrated in Figure 4, the ASR rises sharply to saturate at $k \approx 10$ and maintains a wide stability plateau across $k \in [10, 50]$, with only minor degradation observed at extremes due to insufficient recall (small k) or excessive noise injection (large k). These results indicate that CFA^2 is relatively insensitive to the choice of k within a broad range, with $k = 20$ providing a favorable trade-off between feature coverage and intervention stability.

I Broader Impact

This paper investigates vulnerabilities in large language models (LLMs) to advance safety alignment practices. By exposing these mechanistic weaknesses in widely deployed models, we aim to accelerate the development of robust protective measures before such vulnerabilities can be exploited at scale. While we acknowledge the potential dual-use risks of releasing attack methodologies, we believe

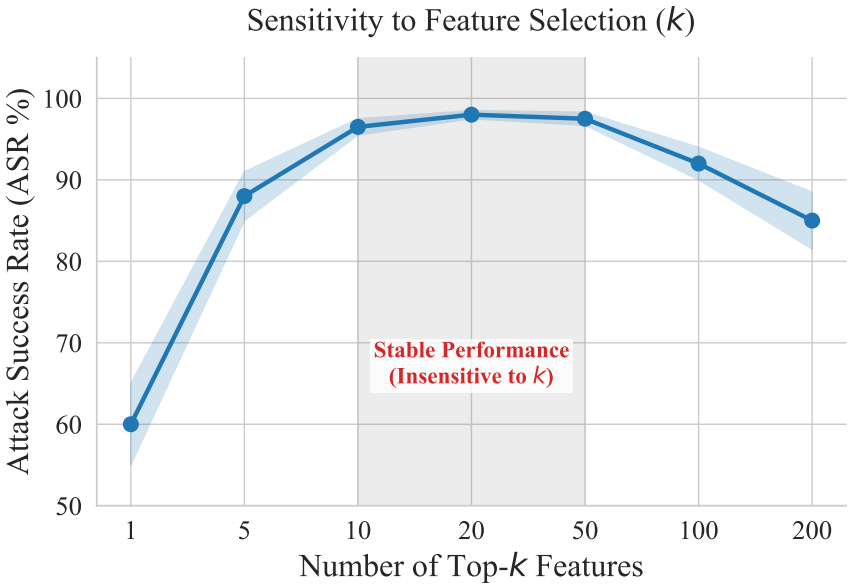

Figure 4: **Sensitivity to Feature Selection.** Attack success rate of CFA^2 under different numbers of Top-k SAE features. Performance remains stable across a broad range of k, particularly for $k \in [10, 50]$.

that transparent, mechanistic research is indispensable for constructing rigorous safeguards and understanding the inner workings of black-box models.