Latent-space Attacks for Refusal Evasion in Language Models

Giorgio Piras[*]
University of Cagliari
giorgio.piras@unica.it

Raffaele Mura[*]
University of Cagliari
raffaele.mura@unica.it

Fabio Brau[*]
University of Cagliari
fabio.brau@unica.it

Maura Pintor
University of Cagliari
maura.pintor@unica.it

Luca Oneto
University of Genova
luca.oneto@unige.it

Fabio Roli
University of Genova
fabio.roli@unige.it

Battista Biggio
University of Cagliari
battista.biggio@unica.it

Abstract

Safety-aligned language models are trained to refuse harmful requests, yet refusal behavior can be suppressed by steering their internal representations. Existing methods do so by ablating a refusal direction from model activations, aiming to remove refusal from the model's residual stream. Despite their empirical success, these methods lack a principled account of the latent-space transformation they induce and why it suppresses refusal. In this work, we recast refusal suppression as a latent-space evasion attack against linear probes trained to separate refused from answered prompts. Under this view, prior work's difference-in-means direction naturally defines such a probe, and its ablation is exactly a projection onto its decision boundary, i.e., a *minimum-confidence* evasion attack. This perspective not only explains the empirical success of prior work but also admits a key limitation: evasion stops at the decision boundary, motivating the need to push representations further into the compliant region, i.e., where the model answers. We leverage this by proposing a *Controlled Latent-space Evasion* attack that projects representations past the boundary with an optimized confidence. Across 15 instruction-tuned, multimodal, and reasoning language models, we achieve higher attack success rates than existing refusal-ablation baselines, also outperforming jailbreak attacks.

Page Code

1 Introduction

Large Language Models (LLMs) are trained on broad web corpora that inevitably include harmful and unsafe content. This makes safety alignment a central requirement, as it allows models to *refuse* harmful requests [1, 2]. However, a growing literature shows that refusal behavior can be manipulated. First, jailbreak attacks demonstrated that carefully crafted input prompts can bypass refusal [3, 4]. More recently, work at the intersection of interpretability and alignment has shown that steering interventions on internal model representations can suppress refusal and enable harmful generations.

[*]These authors contributed equally to this work

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In particular, Arditi et al. [5] show that refusal behavior can be suppressed by a particular steering intervention, named *ablation*, leveraging a direction computed as the difference-in-means (DiM) between activations of harmful and harmless prompts. Ablating this direction consists of projecting each activation onto the hyperplane orthogonal to the DiM. The motivation is intuitive: if refusal is encoded in a single direction, writing it out of the residual stream should prevent the model from refusing. Follow-up work refined this view by selecting directions from orthonormal bases [6], optimizing them via string losses [7], or extracting multiple directions [8]. However, existing work has primarily focused on refining *which* directions to ablate, while implicitly assuming that ablation itself is sufficient to suppress refusal. This lack of principled analysis leaves unanswered *why* ablation is the right intervention, whether it is optimal, and how it can be systematically improved.

This work addresses this gap by providing a principled recasting of refusal suppression as a *latent-space evasion attack* against linear probes. Under this view, suppressing refusal amounts to modifying internal representations so that a linear probe trained to separate harmful from harmless prompts is fooled into classifying a harmful input as harmless. This reframing explains why DiM ablation works. The DiM direction naturally corresponds to the weight vector of a probe, and its ablation is exactly the orthogonal projection of activations onto the probe's decision boundary. This coincides with the DeepFool *minimum-confidence* attack against a binary classifier [9]. Furthermore, this exposes a central limitation of prior work, which projects activations onto the boundary without optimizing the probe's confidence across layers. Since minimum-confidence evasion is, by construction, the weakest perturbation that flips a linear decision, pushing representations past the boundary with explicit confidence control can yield stronger refusal suppression.

Building on this, we propose *Controlled Latent-space Evasion* (CLE), a refusal-suppression mechanism built on a set of linear probes trained at each layer to separate harmful from harmless representations. CLE perturbs activations to evade these probes with a controlled, optimized confidence margin. Since our formulation recasts the goal from erasing refusal (i.e., ablating all activations) to evading the probe, we instantiate CLE in two variants that differ in how the intervention is applied: CLE-P (*projective*) reprojects every token activation, similar to prior work, yet past the probe's boundary by enforcing the target confidence on each projected activation; CLE-A (*additive*) instead, computes the perturbation *once* on the post-instruction token and adds it uniformly to all activations. By comparing these variants, we investigate whether controlled-confidence evasion requires continuous reprojection during generation. We find that a single fixed perturbation is sufficient to sustain evasion, achieving comparable or higher refusal suppression than per-token reprojection.

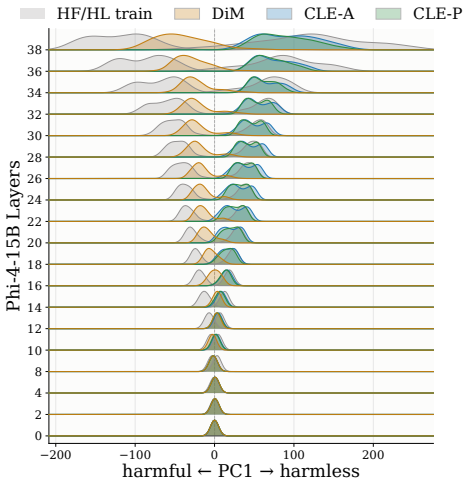


Figure 1: 1st PC of prompt activation across layers: CLE variants confidently shift test prompts into the harmless, compliant region, while DiM leaves activation distribution nearly unchanged.

Fig. 1 visualizes the overall contrast: the minimum-confidence intervention of DiM's minimum-confidence ablation barely shifts harmful representations out of the refusal region, whereas CLE-P and CLE-A project them confidently into the compliant region occupied by harmless prompts. We present our latent-evasion formulation and recasting of prior work in Sect. 2, then describe our CLE-P and CLE-A methods in Sect. 3. We evaluate our approach in Sect. 4 across 15 instruction-tuned, reasoning, and multimodal models, achieving state-of-the-art attack success, demonstrating the effectiveness of controlled latent-space evasion and observing its effect from a mechanistic viewpoint.

2 Refusal Suppression as a Latent-space Evasion Attack

We recast the problem of suppressing refusal in a safety-aligned model as a *latent-space evasion attack* against linear probes defined over internal representations. Sect. 2.1 formalizes refusal suppression in

terms of an abstract oracle that decides whether the model refuses a prompt and motivates surrogating it with a probe over internal activations. Sect. 2.2 shows how this reduces to a latent-space evasion, and serves to both recast prior work in Sect. 2.2.1, and later introduce our approach in Sect. 3.

2.1 Refusal Suppression

We consider an autoregressive language model f composed of L layers $f^{(1)}, \dots, f^{(L)}$, each operating on sequences of d-dimensional token representations, and a last layer $f^{(L+1)}$ providing the next-token scores. We use bold symbols to denote sequences of arbitrary length, i.e., $\boldsymbol{x} \in \left(\mathbb{R}^d\right)^*$ is a sequence in $\mathbb{R}^d$, with $\boldsymbol{x}[i] \in \mathbb{R}^d$ its i-th entry and $\boldsymbol{x}[-1] \in \mathbb{R}^d$ its last entry. We reserve the non-bold symbol for the last-token slice, $x \coloneqq \boldsymbol{x}[-1] \in \mathbb{R}^d$. The activation at the output of the l-th layer for input $\boldsymbol{t}$ is then $\boldsymbol{h}_l(\boldsymbol{t}) \coloneqq \left(f^{(l)} \circ \cdots \circ f^{(1)}\right)(\boldsymbol{t}) \in \left(\mathbb{R}^d\right)^*$, and we denote the *post-instruction* (last-token) representation by $h_l(\boldsymbol{t}) \coloneqq \boldsymbol{h}_l(\boldsymbol{t})[-1] \in \mathbb{R}^d$.

Refusal Oracle. For a prompt $\boldsymbol{t}$, we assume the existence of an *oracle* $\mathcal{Y}$ that provides a binary outcome $\mathcal{Y}(f, \boldsymbol{t})$, which is 0 if and only if the model f fulfills the request in $\boldsymbol{t}$ (i.e., does not refuse to fulfill the request and does not produce a degenerate completion). The oracle is intentionally left abstract: it may be instantiated as an LLM-as-a-judge evaluating the autoregressive completion of f on $\boldsymbol{t}$, $\mathcal{Y}(f, \boldsymbol{t}) \coloneqq \mathcal{J}\big(\boldsymbol{t}, \boldsymbol{o}(\boldsymbol{t})\big)$ with $\boldsymbol{o}(\boldsymbol{t})$ obtained by iterating $\boldsymbol{o}[j+1] = \texttt{next}(f(\boldsymbol{t} \oplus \boldsymbol{o}[:j]))$ (where `next` is any selection of the next token of the autoregressive-based generation) or as any other refusal detector. This abstraction keeps the formulation independent of the specific evaluation protocol.

Refusal-Suppression Problem. Let $\mathcal{D}$ be a distribution of prompts, including both *harmless* and *harmful prompts*, on which the safety-aligned model f is trained to refuse. The attacker's goal is to construct a new, perturbed model Ψf minimizing the following problem:

$$\min_{\Psi} \quad \mathbb{E}_{\boldsymbol{t} \sim \mathcal{D}} \left[\mathcal{Y}(\Psi f, \boldsymbol{t})\right] \tag{1}$$

where Ψ is any transformation applied to either model weights (e.g., with fine-tuning) or by intervening on activations (e.g., through *steering*). In our work, we focus on the latter case by defining steering operators Ψ that modify the residual stream of f at runtime while leaving the weights unchanged; this family encompasses prior refusal-ablation methods [5–8]. Importantly, solving Eq. 1 via direct optimization of $\mathcal{Y}$ is impractical for two reasons: (i) (*Black-box Oracle*) when $\mathcal{Y}$ is instantiated by an external LLM-as-a-judge, gradients with respect to the model's parameters or activations are simply unavailable (ii) (*Computational Cost*) even assuming a local oracle the autoregressive structure of $\mathcal{Y}$ requires backpropagation through repeated next-token sampling, which is computationally prohibitive. We therefore replace $\mathcal{Y}$ by a tractable, gradient-friendly surrogate.

Oracle Surrogate. Observe that the oracle's $\mathcal{Y}$ verdict on $(f, \boldsymbol{t})$ is determined by the activations of the inner layers $\boldsymbol{h}_l(\boldsymbol{t})$ for any prompt $\boldsymbol{t}$, since these deterministically provide the autoregressive completion (more details in App. C). This justifies replacing the expensive or intractable oracle $\mathcal{Y}$ by a surrogate function $\phi\colon \prod_{l=1}^{L} (\mathbb{R}^d)^* \to \mathbb{R}$. Practically, we can reduce to an ensemble ϕ of per-layer *linear probes* each applied on the *post-instruction token* defined as follows:

$$\phi(\boldsymbol{z}_1, \dots, \boldsymbol{z}_L) \coloneqq \sum_{l=1}^{L} \phi_l(\boldsymbol{z}_l) \quad \text{where} \quad \phi_l(\boldsymbol{z}_l) \coloneqq w_l^\top z_l + b_l, \quad \boldsymbol{z}_l \in \left(\mathbb{R}^d\right)^*, \tag{2}$$

where z_l is a short notation for $\boldsymbol{z}_l[-1]$. Crucially, this provides a tractable behavioral proxy for $\mathcal{Y}$: the layer-wise probes approximate refusal decisions from labeled post-instruction residual-stream activations. This choice is motivated by evidence that refusal is linearly represented across layers and models [5, 10], while the construction of the labels and probes is detailed in App. C. For each layer l, the probe ϕ_l outputs a refusal logit for the input activation: positive predicts refusal, negative predicts compliance. Finally, observe that the hyperplane $\mathcal{H}_l = \{x \in \mathbb{R}^d : w_l^\top x + b_l = 0\}$ separates the two regions in latent space, where the weight vectors w_l and biases b_l define per-layer decision boundaries. Henceforth, we assume that w_l has unit norm.

2.2 Latent-space Refusal Evasion

We restrict to transformations Ψ defined by steering interventions on the residual stream, in line with prior work on activation steering for refusal suppression. Hence, we parameterize Ψ with L maps

$\Psi_i \colon \mathbb{R}^d \to \mathbb{R}^d$ inserted between layer blocks as follows:

$$(\Psi f) := f^{(L+1)} \circ \Psi_L \circ f^{(L)} \circ \cdots \circ \Psi_1 \circ f^{(1)}, \tag{3}$$

where each Ψ_i is applied token-wise, i.e., uniformly across token positions. Without loss of generality, let us consider the problem of minimizing $\mathcal{Y}(\Psi f, \boldsymbol{t})$ on a given input prompt $\boldsymbol{t}$, thereby assuming a prompt-dependent steering, which results in the existence of prompt-dependent latent-perturbations $\boldsymbol{\delta}_l(\boldsymbol{t}) \in (\mathbb{R}^d)^*$ such that the steered activation $\tilde{\boldsymbol{h}}_i(\boldsymbol{t})$ can be written in terms of an additive perturbation $\tilde{\boldsymbol{h}}_l(\boldsymbol{t}) = f^{(l)}(\tilde{\boldsymbol{h}}_{l-1}(\boldsymbol{t})) + \boldsymbol{\delta}_l(\boldsymbol{t})$ for each $\boldsymbol{t}$. Problem 1 reduces to the following minimization problem:

$$\min_{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L} \mathcal{Y}(\Psi f, \boldsymbol{t}) \quad \text{s.t.} \quad \tilde{\boldsymbol{h}}_0(\boldsymbol{t}) = \boldsymbol{t}, \quad \tilde{\boldsymbol{h}}_l(\boldsymbol{t}) = f^{(l)}(\tilde{\boldsymbol{h}}_{l-1}(\boldsymbol{t})) + \boldsymbol{\delta}_l(\boldsymbol{t}), \quad l \leq L \tag{4}$$

which corresponds to a *latent-space evasion problem*.

Solving the Refusal Evasion Problem. A sub-optimal solution to Eq. 4 is given by a descending-gradient-based strategy leveraging the surrogate ϕ to estimate the gradient of $\mathcal{Y}$ on $\boldsymbol{\delta}_l$, as follows:

$$\frac{\partial \phi}{\partial \boldsymbol{\delta}_l} = \underbrace{\frac{\partial \phi_l}{\partial \boldsymbol{z}_l}\bigg|_{\boldsymbol{z}_l = \tilde{\boldsymbol{h}}_l(\boldsymbol{t})}}_{\text{direct}} + \underbrace{\sum_{i>l} \frac{\partial \phi_i}{\partial \boldsymbol{z}_i} \frac{\partial f^{(i)}}{\partial \tilde{\boldsymbol{h}}_{i-1}} \cdots \frac{\partial f^{(l+1)}}{\partial \tilde{\boldsymbol{h}}_l}}_{\text{indirect (through subsequent layers)}}. \tag{5}$$

where, for each layer l, the gradient decomposes into a direct $l-$th contribution and an indirect one propagated through the subsequent layers. To avoid backpropagating through model layers, we approximate the full gradient by its direct component, thereby defining a truncated gradient $\hat{g}_l = \nabla_{\boldsymbol{z}_l} \phi_l(\boldsymbol{h}_l)$. Note that, when ϕ_l is instantiated as the linear probe of Eq. 2, this reduces to $\hat{g}_l = [0, \ldots, 0, w_l]$, which is constant, i.e., independent of the input prompt $\boldsymbol{t}$. Following this truncated direction, the sub-optimal solution yields a layer-wise activation update of the form:

$$\boldsymbol{\delta}_l(\boldsymbol{t}) = [0, \ldots, 0, -\alpha_l(\boldsymbol{t})\, w_l] \in \left(\mathbb{R}^d\right)^*, \tag{6}$$

which particularly applies to the post-instruction token, and where $\alpha_l(\boldsymbol{t}) \in \mathbb{R}$ is the step size. In conclusion, Eq. 6 does not merely define a generic update; it rather defines an evasion step against the probe ϕ_l, moving the activation in the direction that increases the probe's compliance confidence.

2.2.1 Recasting Prior Work

We show that the proposed latent-space evasion attack formulation encompasses prior work on ablation-based refusal suppression [5–8], showing how the ablation corresponds to a minimum-confidence evasion attack in latent space. Let us first observe that an interesting choice of the step size is given by $\alpha_l(\boldsymbol{t}) = \phi_l(\boldsymbol{h}_l(\boldsymbol{t}))$, which produces the update

$$\tilde{h}_l(\boldsymbol{t}) = h_l - \phi_l(\boldsymbol{h}_l) \nabla_{z_l} \phi_l(\boldsymbol{h}_l), \tag{7}$$

that coincides with the ℓ_2 projection of h_l into the hyperplane $\mathcal{H}_l$, the classification boundary of ϕ_l. Indeed, because ϕ_l is linear, $\nabla_{z_l} \phi_l(\boldsymbol{h}_l)$ is constant, of unitary norm, and normal to $\mathcal{H}_l$; hence, moving along the gradient with step $\phi_l(\boldsymbol{h}_l)$ traces the shortest path from h_l to $\mathcal{H}_l$ (refer to [9] for the linear case).

Difference in Means (DiM) [5] defines a refusal direction r computed as the difference-in-means between harmful ($\mathcal{D}_{\texttt{hf}}$) and harmless ($\mathcal{D}_{\texttt{hl}}$) activations at a given layer l^*. This corresponds to considering the same probe on all layers, $\phi_l(\boldsymbol{z}_l) = \hat{r}^\top z_l$, with $\hat{r}$ the normalized version of $r = \mu_{l^*} - \nu_{l^*}$, where $\mu_{l^*} := \mathbb{E}_{\mathcal{D}_{\texttt{hf}}} h_{l^*}(\boldsymbol{t})$ and $\nu_{l^*} := \mathbb{E}_{\mathcal{D}_{\texttt{hl}}} h_{l^*}(\boldsymbol{t})$ are the centroids of harmful and harmless representations. The truncated gradient reduces to $\hat{g}_l = \hat{r}$ for all layers $1 \leq l \leq L$, from which the update in Eq. 7 simplifies to $\tilde{h}_l(\boldsymbol{t}) = h_l(\boldsymbol{t}) - (h_l(\boldsymbol{t})^\top \hat{r})\hat{r}$, which strictly coincides with the ablation introduced in the method and, thus, represents a *minimum-confidence evasion* attack.

Refusal Direction Optimization (RDO) [6] extends the DiM-based ablation by refining the choice of the refusal direction r and by also considering an *additive* intervention $h \mapsto h - \alpha\, \hat{r}$ (initially proposed in [11]), which represents a special case of Eq. 6, under a shared probe $\phi_l(\boldsymbol{z}_l) = \hat{r}^\top z_l$, where the perturbation has a constant step size $\alpha_l(\boldsymbol{t}) \equiv \alpha$ independent of $\boldsymbol{t}$.

Promotion Steering (PS) [7] proposes a joint update of the form $h \mapsto x - \hat{r}\hat{r}^\top x + r$. Instantiating the shared probe $\phi_l(\boldsymbol{z}_l) = \hat{r}^\top z_l$, this update can be rewritten as $h \mapsto h - \left(\hat{r}^\top h - \|r\|\right)\hat{r}$, which fits

the general form of Eq. 6 with prompt-dependent step size $\alpha_l(\boldsymbol{t}) = \phi_l(\boldsymbol{h}_l(\boldsymbol{t})) - \|r\|$. Within our framework, this corresponds to a *non-evasive* intervention, as the step size is strictly smaller than the projection step $\phi_l(\boldsymbol{h}_l(\boldsymbol{t}))$, meaning that the steered activation falls short of the decision boundary $\mathcal{H}_l$ remaining in the refusal region at distance $\|r\|$ from the boundary itself.

Multiple Directions (MD) [8], finally, differs from the other approaches in that the update of Eq. 7 is applied through multiple linear probes $\phi_l^{(i)}(\boldsymbol{z}_l) = w^{(i)\top} z_l$ for $i \leq K$, where the directions $\{w^{(i)}\}$ are obtained from the neurons of a self-organizing map (SOM). Using the notation of the paper, the per-direction ablation $\Pi_{w^{(i)}}$—i.e., Eq. 7 with $\alpha_l(\boldsymbol{t}) = \phi_l^{(i)}(\boldsymbol{h}_l(\boldsymbol{t}))$—is applied K times (one for each probe), yielding the shared intervention $\Psi = \Pi_{w^{(1)}} \circ \cdots \circ \Pi_{w^{(K)}}$.

3 Controlled Latent-space Evasion Attacks

In Sect. 2, we recast refusal suppression as a latent-space evasion attack against linear probes. We show that the widely known refusal ablation corresponds to the minimum-confidence update of Eq. 7, which moves the activation onto the decision boundary $\mathcal{H}_l$ and represents the weakest evading perturbation. We leverage this observation here to design a stronger attack. We formalize controlled-confidence evasion in Sect. 3.1, then describe two variants—CLE-P and CLE-A —that differ in how the perturbation is applied across token activations in Sect. 3.2.

3.1 From Minimum-Confidence to Controlled Evasion

The minimum-confidence update of Eq. 7 places every steered activation exactly on $\mathcal{H}_l$—the region of maximum uncertainty—which empirically leads to inconsistent evasion across layers, as shown in Fig. 1. Moving activations past $\mathcal{H}_l$, into the compliant half-space, requires replacing the fixed step size with one that encodes an explicit evasion margin. Fig. 2 provides a graphical interpretation of this idea. Practically, we introduce a per-layer margin $m_l \geq 0$ that controls how far past $\mathcal{H}_l$ each activation is projected, with $m_l = 0$ reducing to a boundary projection. Notably, each margin m_l is a direct measure of the classification confidence of a linear probe (refer to App. E for more details). We refer to this family of evasion attacks as *Controlled Latent-space Evasion*, and operate the resulting confidence control through the choice of the step size, which we now formalize in the following paragraph.

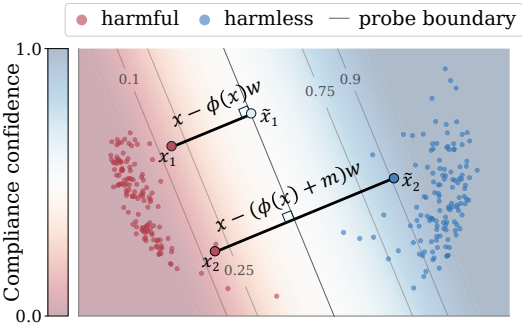


Figure 2: Minimum vs. Controlled evasion in the PCA-rendered latent space of `LLaMA2-7B`.

The Choice of the Step Size. Leveraging latent-space probes as surrogates, the analysis of Sect. 2.1 reduces arbitrary layer-wise perturbations $\boldsymbol{\delta}_l \in (\mathbb{R}^d)^*$ from Eq. 4 to the one-dimensional $\boldsymbol{\delta}_l(\boldsymbol{t}) = [0, \ldots, 0, -\alpha_l(\boldsymbol{t})w_l]$ aligned with the probe's truncated gradient (Eq. 6). With this formulation, controlling evasion confidence amounts to choosing the step α_l that we parameterize as:

$$\alpha_l(\boldsymbol{t}) \;=\; \lambda_l \cdot \big(\phi_l(f^{(l)}(\tilde{\boldsymbol{h}}_{l-1})) + m_l\big), \qquad \lambda_l \in \{0,1\}, \quad m_l \geq 0, \tag{8}$$

where λ_l selects whether layer l is actively steered. In practice, if $\lambda_l = 0$, the layer is left untouched; if $\lambda_l = 1$, it is actively steered with m_l. This allows us to avoid steering on layers where the intervention would be practically useless or would destroy model coherence (see Sect. D.3 for more details). Plugging Eq. 8 into Eq. 4 collapses the optimization problem of Eq. 1 to the search of $2L$ scalars and yields:

$$\min_{\{\lambda_l, m_l\}_l} \quad \mathbb{E}_{\boldsymbol{t}\sim\mathcal{D}}\big[\,\mathcal{Y}(\Psi f,\, \boldsymbol{t})\big] \tag{9a}$$

$$\text{s.t.} \quad \tilde{\boldsymbol{h}}_0(\boldsymbol{t}) = \boldsymbol{t}, \;\; \tilde{\boldsymbol{h}}_l(\boldsymbol{t}) = f^{(l)}\big(\tilde{\boldsymbol{h}}_{l-1}(\boldsymbol{t})\big) + \boldsymbol{\delta}_l, \;\; \forall l, \tag{9b}$$

$$\boldsymbol{\delta}_l = [0, \ldots, 0, -\alpha_l(\boldsymbol{t})w_l], \;\; \alpha_l(\boldsymbol{t}) = \lambda_l\big(\phi_l(f^{(l)}(\tilde{\boldsymbol{h}}_{l-1})) + m_l\big), \tag{9c}$$

$$\lambda_l \in \{0,1\}, \;\; m_l \geq 0. \tag{9d}$$

Algorithm 1 Controlled Latent-space Evasion

Input: Model f with layers $1, \dots, L$; $\mathcal{D}_{\text{train}}$; $\mathcal{D}_{\text{val}}$; judge $\mathcal{Y}_{\text{val}}$; MODE $\in \{\text{CLE-P}, \text{CLE-A}\}$
Output: Steered model Ψf

1: **for** $l = 1, \dots L$ **do**
2: $\quad w_l, b_l \leftarrow \texttt{SVM.fit}(\{(\boldsymbol{h}_l(\boldsymbol{t})[-1], \mathcal{Y}(f, \boldsymbol{t})) : \boldsymbol{t} \in \mathcal{D}_{\text{train}}\})$ ▷ `Fit probes`
3: $\quad w_l \leftarrow \frac{w_l}{\|w_l\|_2}, b_l \leftarrow \frac{b_l}{\|w_l\|_2}$ ▷ `Normalize weights`
4: $\quad$ **if** MODE == CLE-P **then**
5: $\quad\quad$ **def** $\Psi_l(h) : h - \lambda_l\big(\phi_l(h) + m_l\big)w_l$ ▷ `Project every token`
6: $\quad$ **else if** MODE == CLE-A **then**
7: $\quad\quad \delta_l(\boldsymbol{t}) \leftarrow -\lambda_l\big(\phi_l(\boldsymbol{h}_l(\boldsymbol{t})[-1]) + m_l\big)w_l$ ▷ `Compute perturbation`
8: $\quad\quad$ **def** $\Psi(h) : h + \delta_l(\boldsymbol{t})$ ▷ `Apply uniform perturbation`
9: $\quad$ **end if**
10: **end for**
11: $\{\lambda_l^\star, m_l^\star\} \leftarrow \text{BO}\big(\{\lambda_l, m_l\}_l; \Psi f, \mathcal{D}_{\text{val}}, \mathcal{Y}_{\text{val}}\big)$ ▷ `Select layers and margins`
12: **return** $\Psi f = f^{(L+1)} \circ \Psi_L \circ f^{(L)} \circ \cdots \circ \Psi_1 \circ f^{(1)}$

The use of latent-space probes as surrogates reduces the search space from tensors $\boldsymbol{\delta}_l \in (\mathbb{R}^d)^*$ to a set parameterized by a binary layer mask $\{\lambda_l\}$, selecting where evasion matters, and a non-negative margin vector $\{m_l\}$, controlling evasion confidence. In turn, although $\mathcal{Y}$ is black-box and non-differentiable, the resulting minimization problem is amenable to gradient-free black-box optimization on a held-out validation set $\mathcal{D}_{\text{val}}$. We solve Eq. 9 via Bayesian Optimization (BO) [12], and describe the full CLE-P/CLE-A procedure of Algorithm 1 in the following section.

3.2 Instantiating Controlled Evasion: CLE-P and CLE-A

We instantiate CLE in two variants. CLE-P follows prior refusal-ablation methods by reprojecting each token activation, but enforces a target confidence beyond the probe boundary. CLE-A instead computes the evasion perturbation once on the post-instruction token and reuses it across token activations. The two variants are designed to investigate whether controlled-confidence evasion requires continuous reprojection during generation.

Algorithm 1 reports the full procedure, by taking as input the aligned model f, a training $\mathcal{D}_{\texttt{train}}$ and a validation set $\mathcal{D}_{\text{val}}$, a validation oracle $\mathcal{Y}_{\text{val}}$, a mode flag (CLE-P or CLE-A), and by returning a steered model Ψf with optimized layer-wise interventions. For every layer $l \leq L$, we fit the linear probe of Eq. 2 on post-instruction activations from $\mathcal{D}_{\texttt{train}}$, using linear SVMs with normalized weights (Line 3), which define the decision boundaries.

CLE-P (Projective) and CLE-A (Additive) variants. In Line 5, we define the CLE-P intervention as $\Psi_l(h) = h - \lambda_l(\phi_l(h) + m_l)w_l$, applied uniformly to every activation h in both inputs and generated tokens. Geometrically, when $\lambda_l = 1$, the update is the Euclidean projection of h onto the (shifted) hyperplane $\tilde{\mathcal{H}}_l \coloneqq \mathcal{H}_l - m_l w_l$ parallel to $\mathcal{H}_l$ at distance m_l inside the compliant half-space; when $\lambda_l = 0$, Ψ_l collapses to the identity. CLE-P thus generalizes the boundary projection of prior refusal-ablation methods, where each activation is individually projected to $\tilde{\mathcal{H}}_l$. From Line 6 to Line 8 we define CLE-A, which produces a perturbation that is *universal* across decoding tokens. Concretely, the projection is computed *once* at the post-instruction token $h_l = \boldsymbol{h}_l(\boldsymbol{t})[-1]$, yielding a perturbation $\boldsymbol{\delta}_l(\boldsymbol{t})[i] = -\lambda_l(\phi_l(h_l(\boldsymbol{t})) + m_l)w_l$ for all the token activations. The layer-wise intervention, in fact, is then defined as $\Psi_l(h) = h + \boldsymbol{\delta}_l(\boldsymbol{t})$, where the same $\boldsymbol{\delta}_l(\boldsymbol{t})$ is added to every input and generated token activation without any further projection. The resulting perturbation is therefore sufficient to sustain evasion without continuous reprojection of the residual stream, as shown in Sect. 4.

Layer and Margin Search. With Ψ_l parameterized by the 2 scalars (λ_l, m_l), the steered model Ψf is fully parameterized by the $2L$ variables $\{(\lambda_l, m_l)\}_{l=1}^L$, which are optimized to solve Problem 9 via Bayesian Optimization (BO). Each BO query instantiates Ψf with a candidate configuration of $\{\lambda_l\}_l$ and $\{m_l\}_l$, generates completions on $\mathcal{D}_{\text{val}}$, and scores them with the validation oracle $\mathcal{Y}_{\text{val}}$. The procedure returns the configuration $\{(\lambda_l^*, m_l^*)\}_{l=1}^L$ that maximizes attack success, identifying both *where* to evade and *by how much*. Implementation details of the BO setup are deferred to Sect. 4. The steered model returned in Line 12 is the composition Eq. 3, evaluated with the optimized parameters.

4 Experiments

This section describes the experiments conducted to test our CLE methods. We report the controlled latent evasion result (Sect. 4.1), a mechanistic analysis (Sect. 4.2), and ablation studies (Sect. 4.3). We summarize below the employed setup.

Models and Oracles. We consider 9 safety-aligned instruction-tuned models (of which 1 defended [13]), and 6 multimodal and/or reasoning models. During BO search, we use the LLM-as-a-judge `Mistral-7B-val-cls` from Mazeika et al. [14] as a validation oracle $\mathcal{Y}_{\text{val}}$. Differently, to evaluate the Attack Success Rate (ASR) of a given completion at test-time, we instead average results from 2 runs over different seeds using the `HarmBench-Llama-2-13Bcls` from Mazeika et al. [14]. The detailed list of models is given in App. B.

Dataset. We train the probes, either Linear SVM or DiM, following [5] on 128 harmful instructions from ADVBENCH [3], MALICIOUSINSTRUCT [15] and TDC2023 [16] and on 128 harmless instructions from ALPACA [17]. More details regarding probe training can be found in App. C. During CLE BO search, we use the validation dataset $\mathcal{D}_{\text{val}}$ from Mazeika et al. [14], while, at test-time, all methods are evaluated using the HARMBENCH "standard" prompts.

CLE-P and CLE-A. We follow Algorithm 1, running the $\{\lambda_l\}$ and $\{m_l\}$ BO search on $\mathcal{D}_{\text{val}}$ and $\mathcal{Y}_{\text{val}}$ for $500 - 700$ trials (based on model size) using the Tree-Parzen Estimator sampler [18]. For a consistent comparison, both variants use the same selected intervention window, while their margins are optimized independently. The $\{\lambda_l\}$ are selected from all possible contiguous windows, reducing the search space, while allowing all layers to be selected. The margins $\{m_l\}$, searched in $\{10^{-1}k : k = 1, \ldots, 50\}$ with few model exceptions, are reported in App. D along with additional details on the search algorithm. Furthermore, in Sect. 4.3, we validate the use of our search by performing an ablation study on all CLE components, including layer and margin selection.

Baselines. We compare CLE variants against existing refusal suppression methods and prompt-level jailbreak attacks. We consider DiM [5], RDO [6], PS [7], and MD [8] as refusal suppression baselines. We then compare against prompt-level jailbreak attacks, namely GCG [3] and SAA [4], which optimize adversarial prompts for each harmful instruction rather than intervening in latent space. We initially evaluate baselines using their default configurations, detailed in App. B. We additionally optimize their method-specific hyperparameters under the same validation protocol as CLE, to examine the effect of tuning on the comparison (Sect. B.3). By comparing CLEs against both refusal-suppression methods and prompt-specific jailbreak attacks, we aim to provide a comprehensive picture of their effectiveness compared to state-of-the-art methods.

4.1 Results from CLE-P and CLE-A

We discuss here the ASR results from instruct (Tab. 1) and reasoning/multimodal models (Tab. 2).

CLE-P Outperforms Refusal-ablation Baselines. CLE-P projects every activation with a target confidence. Across both Tab. 1 and Tab. 2, it outperforms all refusal-ablation baselines. The gap is particularly pronounced against the minimum-confidence method of DiM ablation: on `LLaMA2-7B`, CLE-P raises ASR from 1.80 to 87.42; on `Mistral-7B-RR`, from 8.80 to 80.50; on `GPT-OSS-20B`, from 0.60 to 86.79. This pattern extends across instruction-tuned, reasoning, and multimodal models. CLE-P reaches an average ASR of 88.34 across all 15 models, exceeding MD by 25.37 points, DiM by 41.01 points, RDO by 62.78 points, and PS by 79.54 points, demonstrating that controlled-margin latent-space evasion is more effective than standard refusal-ablation methods, particularly validating its empirical superiority against minimum-confidence evasion.

CLE-A is Comparable to CLE-P and Outperforms Jailbreak Attacks. CLE-A reaches similar ASR to CLE-P, often improving over it, while still outperforming all refusal-ablation and jailbreak attack baselines. This shows that a single perturbation $\boldsymbol{\delta}_l(\boldsymbol{t})$, computed once on the post-instruction token and added to every activation, is equally or more effective than per-token projection. We interpret this as empirical evidence that continuous reprojection is less critical for sustaining evasion than the confidence with which representations are pushed beyond the decision boundary. A comparison at matched margins further separates the effect of the intervention mechanism from the choice of

Table 1: The attack success rate (ASR) of CLE methods against refusal-ablation baselines and jailbreaks on instruct models. Best ASR in bold, second-best underlined.

Model	DiM	MD	RDO	PS	GCG	SAA	CLE-P	CLE-A
`LLaMA2-7B`	1.80 ±0.60	59.11 ±1.25	1.25 ±0.00	0.60 ±0.60	32.70 ±1.25	57.90 ±0.60	<u>87.42</u> ±1.25	**94.96** ±0.60
`LLaMA3-8B`	72.32 ±1.25	88.05 ±0.60	32.07 ±0.00	0.00 ±0.00	1.90 ±0.60	<u>91.20</u> ±1.25	89.94 ±0.60	**92.45** ±1.25
`Mistral-7B-RR`	8.80 ±0.60	25.79 ±1.25	1.25 ±0.00	0.00 ±0.00	0.60 ±0.60	1.60 ±1.25	<u>80.50</u> ±0.60	**83.02** ±1.25
`LLaMA3.2-3B`	51.57 ±1.25	77.35 ±0.60	53.45 ±0.00	0.00 ±0.00	9.43 ±0.60	47.79 ±1.25	<u>89.94</u> ±0.60	**94.97** ±1.25
`Mistral-7Bv0.3`	81.76 ±0.60	82.38 ±1.25	61.00 ±0.00	0.00 ±0.00	81.76 ±0.60	<u>90.56</u> ±1.25	87.42 ±0.60	**91.82** ±1.25
`Phi3.5-mini`	81.76 ±1.25	80.50 ±0.60	79.87 ±0.00	28.93 ±0.60	25.20 ±1.25	81.80 ±0.60	<u>85.53</u> ±1.25	**91.20** ±0.60
`Olmo3-7B`	35.84 ±0.60	22.01 ±1.25	4.40 ±0.00	2.51 ±0.60	0.60 ±0.00	69.18 ±1.25	<u>95.60</u> ±0.60	**98.11** ±1.25
`Qwen2.5-32B`	87.42 ±1.25	82.38 ±0.60	6.20 ±0.00	26.41 ±1.25	29.55 ±0.60	89.30 ±0.60	<u>91.19</u> ±0.60	**93.08** ±1.25
`Mixtral-8x7B`	64.15 ±0.60	80.50 ±1.25	77.35 ±0.00	11.32 ±0.60	48.42 ±1.25	83.01 ±1.25	<u>85.53</u> ±0.60	**86.16** ±0.60

Table 2: The attack success rate (ASR) of CLE methods against refusal-ablation baselines and jailbreaks on reasoning (**R**) and multimodal (**MM**) models. Best ASR in bold, second-best underlined.

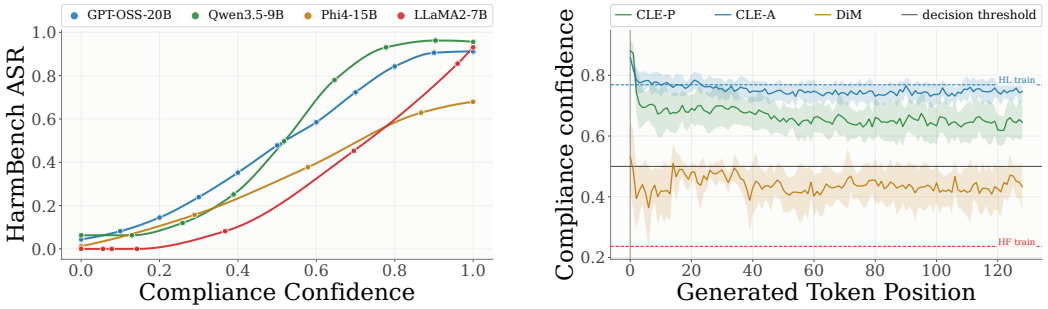
Model	R	MM	DiM	MD	RDO	PS	CLE-P	CLE-A
`GPT-OSS-20B`	✓	✗	0.60 ±0.60	52.20 ±1.25	0.00 ±0.00	0.00 ±0.00	<u>86.79</u> ±0.60	**95.60** ±1.25
`DeepSeek-R1-8B`	✓	✗	57.23 ±1.25	78.61 ±0.60	40.25 ±0.00	0.00 ±0.00	<u>84.91</u> ±1.25	**86.79** ±0.60
`Qwen3.5-9B`	✓	✓	3.31 ±0.60	10.06 ±1.25	1.25 ±0.00	3.77 ±0.60	<u>92.45</u> ±0.00	**97.48** ±1.25
`Phi-4-15B`	✓	✓	0.60 ±0.00	57.23 ±1.25	0.60 ±0.60	0.60 ±0.00	<u>75.47</u> ±0.60	**78.61** ±1.25
`Gemma3-12B`	✗	✓	66.66 ±1.25	53.45 ±0.60	14.46 ±0.00	7.54 ±1.25	<u>95.60</u> ±0.60	**100.00** ±0.00
`Ministral3-14B`	✗	✓	96.22 ±0.60	94.96 ±1.25	10.06 ±0.00	50.31 ±1.25	<u>96.85</u> ±0.00	**99.37** ±0.60

evasion confidence (Sect. G.1). Beyond refusal-ablation baselines, we compare CLEs against two prompt-level jailbreak attacks, GCG and SAA. Crucially, jailbreak attacks differ as they operate with arguably more favorable settings: they solve a separate optimization problem for every single harmful prompt, while CLE methods optimize (only) $2L$ scalar coefficients per model, therefore reusable for any prompt and thus not requiring prompt-specific tuning. From Tab. 1, CLE-A consistently outperforms jailbreak methods, with margins up to 81 points on `Mistral-7B-RR`. CLE-A thus achieves stronger refusal suppression at a fraction of the cost: $\sim 10^3$ forwards per prompt for GCG/SAA [19], versus a single model forward for CLE.

Robustness to Tuning and Prompt Distribution. For a thorough CLE validation, we must exclude two additional factors that could contribute to the observed gains: the larger hyperparameter-search budget used by CLE, and the selection of its intervention parameters on HARMBENCH. To isolate the role of tuning, we optimize the baseline-specific hyperparameters using the same validation set and judge. On four models selected for their comparatively strong baseline performance, CLE-A retains a higher mean ASR (91.20%) than MD, the strongest baseline on average (79.71%; Sect. B.3). We also test whether the optimized interventions generalize beyond the prompt distribution used for selection. Without further tuning (i.e., using the same best configuration found on HARMBENCH), CLE-A outperforms DiM on `WildJailbreak` [20] and on Italian and Chinese `MultiJail` [21] prompts across 13 models, although its absolute ASR decreases in the multilingual setting (Sect. B.4). These experiments suggest that the improvement is not solely attributable to additional tuning or to the original evaluation distribution.

4.2 Mechanistic Analysis

Confidence Control. Actively controlling the target evasion confidence improves attack success rate. Fig. 1 shows the distribution of post-instruction token activations along the first principal component across the layers of `Phi-4-15B` on the HARMBENCH test set. While DiM ablation barely shifts harmful representations toward the harmless region, CLEs confidently push them into the compliant half-space, closely overlapping with harmless probe-training prompts. The figure also explains why the optimized intervention window is layers 9–30: early layers exhibit heavily overlapping harmful/harmless distributions, providing little useful signal for steering. Fig. 3a confirms this view: CLE-A ASR increases monotonically with *compliance confidence* (defined by $1 - \sigma(\phi(\cdot))$), with near-zero attack success at zero margin (i.e., minimum-confidence) and high ASR at larger margins.



(a) (b)

Figure 3: (a) ASR grows monotonically with the compliance confidence (i.e., $1 - \sigma(\phi(z))$). (b) `Phi-4-15B` probe confidence of generated tokens. CLE-A sustains the evasion just as CLE-P.

Projection vs Addition. Fig. 3b shows `Phi-4-15B`'s layer 36 probe compliance confidence during token generation, where dotted lines represent the average value, showing that an additive perturbation on the activations is sufficient to keep the subsequent generation process in the compliant region, thereby suppressing refusal. This suggests that once the latent trajectory shifts into the compliant region, a fixed perturbation can sustain evasion throughout generation (see further details in App. F).

4.3 Ablating CLE-P and CLE-A Components

In Fig. 4, we isolate the main components of CLE-A by running BO search for each variant. Replacing SVM probes with DiM probes consistently lowers ASR, proving that an accurate oracle surrogate (discussed in App. C) translates into stronger evasion. Similarly, setting $\lambda_l \equiv 1$ (i.e., evading every probe) while optimizing only the margins is also suboptimal: although this keeps confidence control, it removes the ability to select where the intervention should act, often perturbing layers where steering is unnecessary. Additionally, setting $m_l \equiv 0$, while optimizing the λ_l, recovers a zero-margin additive intervention, showing that layer selection alone is not sufficient without positive margin control. Finally, the weakest variant fixes both $\lambda_l \equiv 1$ and $m_l \equiv 0$, corresponding to an all-layer minimum-confidence intervention, achieving the lowest performance. Taken together, these ablations show that CLE-A is not driven by a single design choice, but by the combination of accurate probes, layer selection, and margin optimization. BO is therefore essential to identify both a compact intervention window and the amount of confidence required at each layer. We report analogous results for CLE-P in App. G and complement this analysis with model-coherence experiments in App. H, quantifying how refusal suppression can trade off against general task accuracy.

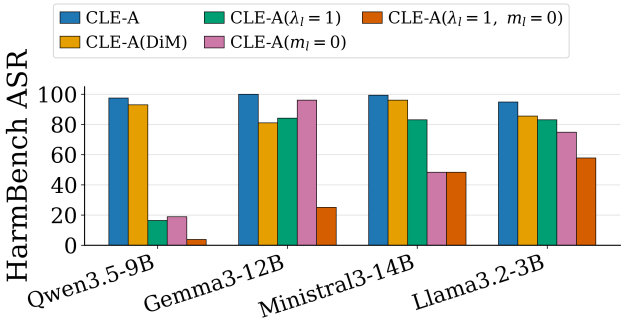


Figure 4: Ablation study among CLE components.

5 Related Work

Our work studies the limits of LLM safety alignment from a mechanistic perspective. It differs from prompt-level jailbreaks such as GCG [3] and SAA [4], which optimize input-space suffixes for each harmful prompt. Instead, we recast activation steering as latent-space evasion and intervene directly on internal representations; for completeness, we compare against these jailbreak attacks and consistently outperform them. Closest to our work are refusal-direction ablation methods, already discussed throughout the work. Arditi et al. [5] first showed that refusal can be suppressed by ablating a DiM

direction from the residual stream, following also the hypothesis from Park et al. [10]. Subsequent work refined this direction by modeling refusal through cone-like orthogonal bases [6], optimizing steering directions with string-based losses [7], or extracting multiple directions with Self-Organizing Maps [8], in line with recent evidence that concepts may have richer geometric structure [22–24]. We reframe these methods as latent-space evasion attacks and show that our controlled variants, CLE-P and CLE-A, outperform them. Operationally, CLE-P extends ablation by replacing boundary projection with controlled-confidence evasion, while CLE-A relates to activation-addition methods. Prior additive approaches either add a prompt-independent DiM shift [5] or contrastive activation vectors [25, 11]. In contrast, CLE-A computes a prompt-dependent perturbation on the post-instruction token that evades the probe with controlled confidence and then sustains this evasion without reprojection.

6 Conclusions, Limitations and Future Work

We recast refusal suppression in safety-aligned language models as a latent-space evasion attack against linear probes separating refused from compliant representations. Under this view, prior refusal-ablation methods reduce to minimum-confidence evasion, which is, by construction, the weakest perturbation flipping a linear decision. Based on this, we proposed *Controlled Latent-space Evasion* (CLE), which pushes representations past the boundary with an optimized confidence margin, and instantiated it in two variants: CLE-P, reprojecting every token activation during generation, and CLE-A, computing the perturbation once on the post-instruction token and applying it universally. Empirically, both CLE-A and CLE-P generally achieve higher ASR than existing refusal-ablation baselines and jailbreak attacks. A first *limitation* of this work is that both CLE methods assume white-box access to internal activations and the ability to intervene in the residual stream; it therefore does not directly apply to black-box model APIs. Moreover, always-on steering can reduce general task performance. A further limitation of the work is the use of BO to select layers and margins. Future work can be oriented to parameterize λ_l, m_l as differentiable variables and fold them into the gradient-based formulation of Eq. 4, replacing BO with direct gradient-based optimization. Finally, the CLE approach relies on the linear separability of refused and answered requests in latent space, which consistently holds across all models we tested, enabling reliable evasion. Therefore, alignment procedures that actively discourage linear separability could, in principle, reduce the attack effectiveness. Future work could focus on mitigating such a family of attacks by acting on the separability property at the latent space, e.g., using alignment procedures that actively discourage it by distributing refusal representations across non-trivial manifolds.

Acknowledgements

This work was partly supported by the EU-funded Horizon Europe projects Sec4AI4Sec (GA No. 101120393) and CoEvolution (GA No. 101168560), by project FISA-2023-00128, funded under the MUR program "Fondo Italiano per le Scienze Applicate," and by Fondazione di Sardegna under the project "LatentShield: Protecting Large Language Models from Prompt Injection in Latent Space" (CUP: F83C26000350007).

Competing Interests. The authors declare no competing interests.

References

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

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut

Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[3] 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.

[4] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hXA8wqRdyV.

[5] Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pH3XAQME6c.

[6] Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=80IwJqlXs8.

[7] Jacob Dunefsky and Arman Cohan. One-shot optimized steering vectors mediate safety-relevant behaviors in LLMs. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=teW4nIZ1gy.

[8] Giorgio Piras, Raffaele Mura, Fabio Brau, Luca Oneto, Fabio Roli, and Battista Biggio. Som directions are better than one: Multi-directional refusal suppression in language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):32728–32736, March 2026. ISSN 2159-5399. doi: 10.1609/aaai.v40i39.40551. URL http://dx.doi.org/10.1609/aaai.v40i39.40551.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[10] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[11] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

[12] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Neural Information Processing Systems*, volume 2 of *NIPS'12*, page 2951–2959. Curran Associates Inc., 2012.

[13] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=IbIB8SBKFV.

[14] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024.

[15] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source LLMs via exploiting generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=r42tSSCHPh.

[16] Mantas Mazeika, Andy Zou, Norman Mu, Long Phan, Zifan Wang, Chunru Yu, Adam Khoja, Fengqing Jiang, Aidan O'Gara, Ellie Sakhaee, Zhen Xiang, Arezoo Rajabi, Dan Hendrycks, Radha Poovendran, Bo Li, and David Forsyth. Tdc 2023 (llm edition): The trojan detection challenge. In *NeurIPS Competition Track*, 2023.

[17] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[18] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[19] Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long, Kenji Kawaguchi, Anirudh Goyal, and Michael Q Shieh. Accelerating greedy coordinate gradient and general prompt optimization via probe sampling. *Advances in Neural Information Processing Systems*, 37:53710–53731, 2024.

[20] Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 47094–47165, 2024.

[21] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vESNKdEMGp.

[22] Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

[23] Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=CqViN4dQJk.

[24] Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025.

[25] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Supplementary materials of "Latent-space Attacks for Refusal Evasion in Language Models"

Disclaimer: This work contains content that may be considered harmful or offensive.

A Ethical and Societal Impact

This work exposes a weakness of current safety-aligned language models: refusal behavior can be suppressed by targeted latent-space interventions. Similar to prior work on jailbreaks and refusal ablation, this research is dual-use and could be misused to weaken model safeguards, especially in white-box settings where internal activations can be modified. At the same time, studying this failure mode is important for understanding how refusal is represented inside language models and why current alignment mechanisms remain vulnerable. Our contribution is intended to support defensive research by providing a principled account of refusal suppression as latent-space evasion, clarifying the limitations of existing ablation methods, and motivating stronger safety mechanisms that are less dependent on linearly separable refusal features. We therefore believe that the scientific and safety benefits of disclosing these results outweigh the associated risks.

Responsible Release. To support reproducibility while limiting misuse, we release the probe-training code, the intervention implementation, the Bayesian Optimization search procedure, and the evaluation scripts needed to verify our results. We do not release pre-fitted probe weights, packaged attack entry points, or generated completions. This scope applies to the paper as well: we report attack outcomes through judge verdicts and behavioural descriptions rather than reproducing harmful text. The model-specific margins reported in Tab. 6 are log-odds defined with respect to each model's fitted probe. They are therefore not directly executable or transferable without independently fitting the corresponding probes. We retain these configurations in the paper because they are necessary for scientific transparency and reproducibility, while excluding the pre-fitted components and packaged tooling that would make them immediately deployable.

Responsible Use of the Method. Techniques for probing and manipulating latent representations advance our understanding of how safety alignment is encoded, but they carry inherent dual-use risk. We define the intended use of Controlled Latent-space Evasion (CLE) as a resource for defenders, i.e., researchers characterizing refusal mechanisms, strengthening alignment robustness, and developing representation-level defenses. Acceptable uses are strictly limited to research on the latent-space basis of refusal, improving the robustness of safety alignment, secure system design, and proactive red-teaming of models the researchers own or are authorized to evaluate. We explicitly prohibit any offensive application, including circumventing safety mechanisms in deployed third-party systems for eliciting harmful content, or deploying refusal-evasion for malicious purposes. Consistent with our Responsible Release policy above, we release code oriented toward analysis and evaluation rather than as a turnkey bypass tool, and we do not release the harmful completions generated during evaluation.

Safeguards and Mitigation Strategies. CLE operates by modifying internal representations and therefore requires white-box access to a model's activations and weights; this dependency itself delimits the threat model and suggests concrete mitigations. First, model providers should protect internal state: serving APIs should not expose intermediate activations, hidden states, or fine-grained logits that could facilitate reconstruction of the refusal-relevant subspace, and access to open weights should be governed by appropriate release controls. Second, and most directly, our results show that representation-level defenses are necessary but not yet sufficient: CLE bypasses the Representation Rerouting / circuit-breaker defense [13]. This motivates the mitigation direction we identify in our analysis, namely alignment procedures that actively discourage the linear separability CLE exploits, e.g., by distributing refusal representations across non-trivial manifolds rather than a small number of steerable directions, alongside latent-space adversarial training and inference-time activation monitoring. Third, providers releasing open-weight models should pair them with robustness evaluations against representation-manipulation attacks, so that alignment also reflects resilience to latent-space intervention rather than input/output filtering alone.

Ethical Reflection and Transparency. We acknowledge the dual-use nature of this work. Our judgment is that disclosure is net beneficial. The vulnerability we characterize already exists in released open-weight models, and defenders cannot address it without knowing that refusal is linearly separable and therefore evadable with controlled confidence. We also note the marginal risk is limited: an adversary with the weight-level access CLE requires can already suppress refusal by fine-tuning, typically at comparable cost and with less expertise, so the method's contribution is primarily diagnostic—it characterizes *how* refusal is encoded, which is information defenders need. That a dedicated representation-level defense is bypassed instead suggests current alignment claims may overestimate resilience to latent-space intervention, a gap better identified by researchers than discovered by adversaries.

B Experimental Setup

We report here the specific configurations and hyperparameters used for the considered baselines. Unless otherwise specified, all methods are evaluated with the default settings provided by their respective codebases.

B.1 Models and Judges.

We consider 9 safety-aligned instruction-tuned models: `LLaMA2-7B`, `LLaMA3-8B`, `LLaMA3.2-3B`, `Mistral-7Bv0.3`, `Phi3.5-mini`, `Olmo3-7B`, `Qwen2.5-32B`, `Mixtral-8x7B`, and one robust model implementing the Representation Rerouting (RR) defense, `Mistral-7B-RR` [13]. In addition, we consider 6 multimodal and/or reasoning models: `GPT-OSS-20B`, `DeepSeek-R1-8B`, `Qwen3.5-9B`, `Phi-4-15B`, `Gemma3-12B`, and `Ministral3-14B`. During BO search, we use the LLM-as-a-judge `Mistral-7B-val-cls` from [14] as a validation oracle $\mathcal{Y}_{\text{val}}$. Differently, to evaluate the Attack Success Rate (ASR) of a given completion at test-time, we instead average results from 2 runs over different seeds using instead the `HarmBench-Llama-2-13Bcls` from [14].

B.2 Baseline Methods for Refusal Suppression

DiM. We evaluate DiM [5] using the official codebase and its default pipeline. We compute candidate DiM directions from 128 harmful and 128 harmless instructions as detailed in Sect. 4. Directions are obtained for each post-instruction token position and layer. The pipeline selects the best direction by ranking candidates according to their refusal score, steering score, and KL-divergence score. We report the results obtained with the best selected direction.

RDO. We evaluate RDO [6] using the released codebase and default settings. We train with a batch size of 16 and a base learning rate of 0.01 using AdamW, and search over refusal cones with dimensionality between 2 and 3. After generating candidate refusal directions at the same layer as DiM, we report results for the most effective direction.

PS. Following [7], we evaluate Promotion Steering (PS) by optimizing 12 steering vectors from pairs of harmful requests and affirmative responses provided in the official codebase. We optimize and apply each steering vector at the same layer used for the DiM baseline. We use the default configuration, with a learning rate of 0.1 and 40 optimization iterations. Across all models, we report the results obtained with the most effective optimized direction.

MD. Following [8], we evaluate MD with its recommended hyperparameters. We train a 4×4 SOM on the same latent layer used for the DiM baseline for $10{,}000$ iterations, with learning rate 0.01, resulting in 16 candidate refusal directions. We then run Bayesian Optimization with 512 trials to select 7 directions, and report the results obtained by ablating the most effective set.

GCG. We evaluate GCG [3] using the implementation provided by HARMBENCH [14]. Across all models, we use the default configuration, with a modifiable token subset size of $|\mathcal{I}| = 20$, $T = 500$ optimization iterations, and $topk = 256$.

SAA. We evaluate SAA [4] using the default hyperparameters provided in the authors' codebase, namely $T = 10000$ optimization iterations and a modifiable token subset size of $|\mathcal{I}| = 25$. Across all models, we use the fixed "refined best" template.

B.3 Matched-budget Comparison with Refusal-ablation Baselines

The main comparison uses the original configurations of the refusal-suppression baselines. Since CLE selects its layer window and margins through BO, however, part of its advantage could reflect a larger validation-time search rather than the intervention itself. We therefore optimize the baselines' tunable hyperparameters using the same validation set and oracle as CLE, while retaining each method's original formulation. For DiM, we exhaustively evaluate candidate directions from all layers instead of relying on the original direction-selection heuristic. For PS, we evaluate six promotion-steering directions at each layer. For RDO, we jointly optimize the intervention layer and the loss coefficient by BO with the same trial budget as CLE. The MD baseline already uses 512 BO trials, and its optimized result is retained. We report this comparison on four models where the original baselines achieve comparatively high ASR, making them informative cases for assessing whether additional tuning closes the gap. As shown in Tab. 3, CLE-A outperforms every baseline on all four models, with 91.20% mean ASR versus 79.71% for MD, the strongest baseline on average. Exhaustive layer selection does not improve DiM over its original heuristic on these models, whereas the average ASR of RDO increases from 58.64% to 61.25%. The remaining gap is consistent with the component ablations in Sect. 4.3, in which ablated CLE variants receive the same BO procedure and budget. Taken together, these results indicate that the gains from controlled-confidence evasion cannot be explained by the search budget alone.

Table 3: ASR (%) under matched tuning on four representative models. Values for RDO, DiM and PS are tuned / default; MD already uses a 512-trial BO search. The selected models have relatively strong default DiM or RDO performance.

Model	CLE-A	RDO	DiM	PS	MD
`LLaMA3.2-3B`	**94.97**	71.70 / 53.45	51.57 / 51.57	0.00 / 0.00	77.35
`Mistral-7Bv0.3`	**91.82**	59.10 / 61.00	81.76 / 81.76	0.00 / 0.00	82.38
`Phi3.5-mini`	**91.20**	68.30 / 79.87	81.76 / 81.76	51.60 / 28.93	80.50
`DeepSeek-R1-8B`	**86.79**	45.90 / 40.25	57.23 / 57.23	3.80 / 0.00	78.61
Average	**91.20**	61.25 / 58.64	68.08 / 68.08	13.85 / 7.23	79.71

B.4 Generalization to Other Harmful-prompt Distributions

The layer windows and margins of CLE are optimized on the HARMBENCH validation distribution. We therefore investigate whether the resulting intervention also suppresses refusal when harmful requests differ in their wording, framing, or language. We consider 2,000 prompts from WildJailbreak [20], including indirect requests and role-play framing, and 315 Italian and 315 Chinese prompts from MultiJail [21]. For MultiJail, we translate generated responses with `NLLB-200-3.3B` before applying the evaluation judge. For each model, we reuse the probes fitted on English data and the layer windows and margins selected on HARMBENCH. Neither CLE-A nor DiM is retuned on the target distributions, so the evaluation jointly tests transfer of the learned probe directions and the fixed intervention parameters. Tab. 4 reports the ASR of both methods on 13 models. CLE-A exceeds DiM for every model and dataset, with mean ASR of 92.36% versus 57.92% on WildJailbreak, 53.94% versus 17.82% on Italian MultiJail, and 52.04% versus 15.77% on Chinese MultiJail. The lower absolute ASR on MultiJail suggests that cross-lingual evaluation remains challenging, although differences in dataset difficulty and translated-response judging may also contribute.

Table 4: Cross-distribution ASR (%) without additional tuning. The original HarmBench-selected layer windows, margins, and English-fitted probes are reused for all tests. WildJail = WildJailbreak; MJ-it / MJ-zh = MultiJail in Italian / Chinese. Each pair reports CLE-A and DiM.

Model	**WildJail**		**MJ-it**		**MJ-zh**	
	CLE-A	DiM	CLE-A	DiM	CLE-A	DiM
`LLaMA2-7B`	**90.25**	6.15	**60.95**	0.32	**65.71**	0.00
`LLaMA3-8B`	**95.05**	77.55	**57.78**	23.49	**57.46**	16.51
`Mistral-7B-RR`	**85.75**	20.65	**49.84**	0.00	**50.16**	1.90
`LLaMA3.2-3B`	**96.20**	75.30	**60.32**	18.10	**50.16**	15.87
`Mistral-7Bv0.3`	**94.95**	91.95	**53.65**	32.70	**56.19**	35.56
`Phi3.5-mini`	**93.71**	91.00	**65.40**	36.51	**55.24**	35.23
`Olmo3-7B`	**96.15**	62.75	**31.43**	16.83	**46.35**	9.84
`GPT-OSS-20B`	**91.55**	0.45	**56.19**	0.20	**49.21**	0.20
`DeepSeek-R1-8B`	**86.80**	84.35	**39.68**	28.89	**17.46**	10.48
`Qwen3.5-9B`	**92.10**	5.15	**58.73**	4.12	**68.89**	3.18
`Phi-4-15B`	**84.45**	52.65	**45.71**	15.56	**43.17**	17.78
`Gemma3-12B`	**96.50**	90.15	**56.83**	6.67	**51.43**	7.62
`Ministral3-14B`	**97.20**	94.90	**64.76**	48.25	**65.08**	50.80

C Latent-space Probes as Surrogates

C.1 Probe Training and Evaluation

We consider one probe per model layer, either training a Linear SVM or using a difference-in-means (DiM) direction, as introduced in Sect. 2 and further discussed in Sect. 2.2.1. The training set contains 128 harmful instructions sampled from ADVBENCH [3], MALICIOUSINSTRUCT [15], and TDC2023 [16], and 128 harmless instructions sampled from ALPACA [17]. We follow the filtering pipeline of [5]: harmful instructions are removed if the unmodified model already complies with them, while harmless instructions are removed if the model refuses them. This ensures that probe labels are consistent with the model's observed behavior, and, thus, that the probes serve as good surrogates of the judge. Following Sect. 2.1, for each remaining prompt, we extract the post-instruction representation at each layer's output, obtaining one training set per layer. Linear SVM probes are trained independently at each layer using `LinearSVC` with a regularization parameter $C = 0.1$, and a maximum number of iterations set to 10^6. The learned weight vector is normalized before being used in the intervention, matching the unit-norm convention assumed in Sect. 2.

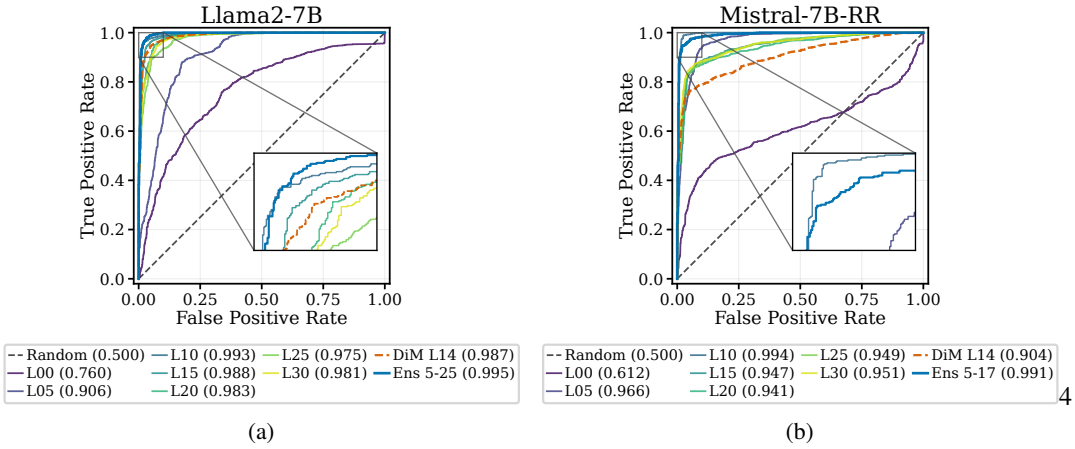
C.2 Why Activations Can Surrogate Refusal

The formulation in Sect. 2.1 replaces the refusal oracle with layer-wise probes over internal activations. This surrogate is useful only if refusal behavior is already predictable from those activations. We validate this assumption on 1000 harmful prompts from SORRYBENCH and 1000 harmless prompts from ALPACA. Following [5], these samples are labeled from the model's actual completions, consistently with the oracle definition in Sect. 2.1: harmful prompts are assigned label 1 when the model refuses them, while harmless prompts are assigned label 0 when the model fulfills the request. The probe is therefore evaluated on its ability to predict, from activation space, the eventual outcome of the model's response.

Fig. 5 reports ROC curves for `LLaMA2-7B` and `Mistral-7B-RR`. Single-layer SVM probes already achieve high AUC in most intermediate and late layers. For `LLaMA2-7B`, layers from 10 onward are almost perfectly separating, while the first layer is substantially weaker. For `Mistral-7B-RR`, the same trend holds: layer 0 is weak, whereas layers 5 and above provide strong linear separation. This shows that the refusal/compliance outcome of a prompt is linearly evident from internal representations before the completion is generated.

The figure also reports the ensemble score defined in Eq. 2, obtained by aggregating the layer-wise probe logits across a contiguous set of predictive layers. This ensemble further improves separation,

reaching near-perfect AUC on both models. Finally, the ROC curves include the DiM probe used in [5]. While DiM is competitive on `LLaMA2-7B`, it is weaker than the SVM ensemble and becomes clearly less reliable on `Mistral-7B-RR`. We analyze this comparison more directly in the next section.



(a) (b)

Figure 5: ROC curves for single-layer SVM probes, the SVM probe ensemble, and the DiM probe on 1000 SORRYBENCH and 1000 ALPACA samples for (a) `LLaMA2-7B` and (b) `Mistral-7B-RR`.

C.3 Difference in Means is not optimal

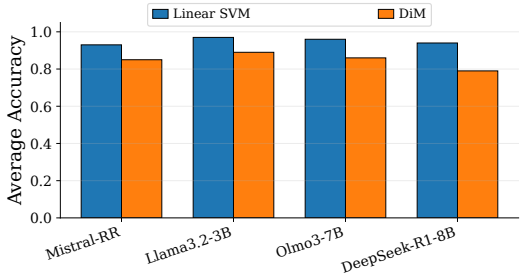


Figure 6: Average layer-wise accuracy of Linear SVM and DiM probes on the HARMBENCH test set.

By construction, the SVM fitted on the data minimizes classification error, and therefore outperforms the DiM probe, which corresponds to a special case of a linear classifier with a suboptimal weight vector. We compare the Linear SVM probes used by CLE-P and CLE-A with the DiM probes in Fig. 6, reporting the average probe accuracy across layers on the HARMBENCH test set. Linear SVM probes consistently outperform DiM probes across the considered models. The gap is moderate on models such as `LLaMA3.2-3B`, but becomes larger on `DeepSeek-R1-8B`, where the DiM is less predictive. This confirms the trend observed in Fig. 5: DiM directions can encode refusal information, but they provide weaker surrogate decisions than SVM probes. This supports using SVM probes as the default surrogate in our controlled-evasion methods, as they provide more accurate and reliable predictions of refusal/fulfillment than DiM probes.

D Bayesian Optimization Algorithm and Results

This section provides the details of the Bayesian Optimization procedure used to select the active layers and margins for CLE-P and CLE-A. As discussed in Sect. 4, the probes are fixed after training, and the search is restricted to the intervention parameters controlling where latent-space evasion is applied and with which target confidence. We first describe the search algorithm, and then report the optimized layer windows and margins used in the experiments.

Algorithm 2 BO search for controlled latent-space evasion

Input: Model f; probes $\{\phi_l\}_{l=1}^{L}$; validation set $\mathcal{D}_{\text{val}}$; judge $\mathcal{Y}_{\text{val}}$; margin grid $\mathcal{M}$; total trials T.
Output: Best layer mask $\{\lambda_l^\star\}_{l=1}^{L}$ and margins $\{m_l^\star\}_{l=1}^{L}$

$T_C \leftarrow \lfloor 0.7T \rfloor$; $\eta \leftarrow 0.1$ (margin-grid step); $\text{ASR}^\star \leftarrow -\infty$
for $t \leftarrow 1$ to T_C **do**
  $(s, e, m) \leftarrow \texttt{BO.suggest}(\{(s, e) : 1 \le s < e \le L + 1\}, \mathcal{M})$
  $\lambda_l \leftarrow \mathbf{1}\{s \le l < e\}, \quad m_l \leftarrow \lambda_l m$, for $l = 1, \ldots, L$
  $\text{ASR} \leftarrow \textsc{Trial}(\{\lambda_l\}, \{m_l\}, f, \{\phi_l\}, \mathcal{D}_{\text{val}}, \mathcal{Y}_{\text{val}})$
  $\texttt{BO.observe}(s, e, m, \text{ASR})$
  if $\text{ASR} > \text{ASR}^\star$ **then**
    $\text{ASR}^\star \leftarrow \text{ASR}$; $\{\lambda_l^\star\} \leftarrow \{\lambda_l\}$; $m^\star \leftarrow m$
  end if
end for
$m_l^\star \leftarrow \lambda_l^\star m^\star$, for $l = 1, \ldots, L$
for $t \leftarrow T_C + 1$ to T **do**
  $m_l \leftarrow 0$, for $l = 1, \ldots, L$
  $m_l \leftarrow \texttt{Sample}(\mathcal{M} \cap \{m^\star + k\eta \mid k = -10, \ldots, 10\})$ for all l with $\lambda_l^\star = 1$
  $\text{ASR} \leftarrow \textsc{Trial}(\{\lambda_l^\star\}, \{m_l\}, f, \{\phi_l\}, \mathcal{D}_{\text{val}}, \mathcal{Y}_{\text{val}})$
  if $\text{ASR} > \text{ASR}^\star$ **then**
    $\text{ASR}^\star \leftarrow \text{ASR}$; $\{m_l^\star\} \leftarrow \{m_l\}$
  end if
end for
return $\{\lambda_l^\star\}_{l=1}^{L}, \{m_l^\star\}_{l=1}^{L}$
function $\textsc{Trial}(\{\lambda_l\}, \{m_l\}, f, \{\phi_l\}, \mathcal{D}_{\text{val}}, \mathcal{Y}_{\text{val}})$
  Instantiate $\Psi_{\{\lambda_l\},\{m_l\}} f$ using Algorithm 1
  Generate $\{\hat{o}(\boldsymbol{t})\}_{\boldsymbol{t} \in \mathcal{D}_{\text{val}}}$ with $\Psi_{\{\lambda_l\},\{m_l\}} f$
  return $\frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{\boldsymbol{t} \in \mathcal{D}_{\text{val}}} \mathbf{1}\{\mathcal{Y}_{\text{val}}(\boldsymbol{t}, \hat{o}(\boldsymbol{t})) = \text{success}\}$
end function

D.1 Bayesian Optimization Algorithm

The Bayesian optimizer searches for the active layer window and target margins that maximize ASR on the HARMBENCH validation set. We outline the procedure in Algorithm 2. The search is parameterized by a contiguous layer window $\{s, \ldots, e-1\}$, inducing the binary layer-selection variables $\lambda_l = \mathbf{1}\{s \le l < e\}$. This reduces the dimensionality of the search space while still allowing the optimizer to select any intervention region in the model.

The first stage uses 70% of the trial budget to search over a layer window (s, e) and a shared margin m. Each candidate configuration is evaluated through the `TRIAL` function, which instantiates the steered model $\Psi_{\{\lambda_l\},\{m_l\}} f$ using Algorithm 1, generates completions on $\mathcal{D}_{\text{val}}$, and computes ASR with the validation oracle $\mathcal{Y}_{\text{val}}$. The sampler is initialized with random trials and then updated with the observed (s, e, m) and ASR values, following a Tree-Parzen Estimator (TPE) strategy.

After this coarse search, the remaining 30% of the budget is used to locally refine the margins on the best selected layer window. The window is kept fixed, and each active layer margin is sampled from the grid spanning $[m^\star - 1, m^\star + 1]$ at increments of $\eta = 0.1$ (intersected with the allowed margin grid). This gives a lightweight per-layer adjustment without expanding the full BO search to all layer-wise margins. The best observed configuration is finally returned as $\{\lambda_l^\star\}_{l=1}^{L}, \{m_l^\star\}_{l=1}^{L}$.

The cost of the procedure is dominated by `TRIAL`, since every trial requires running the steered model on the validation set and evaluating the generated completions with the validation judge. The overall cost therefore scales with the number of trials and the size of $\mathcal{D}_{\text{val}}$. In our experiments, we use $T = 500$ trials for most models, increasing it to 700 for models with more than 32 layers, which we found sufficient for the reduced search space induced by the contiguous layer-window parameterization.

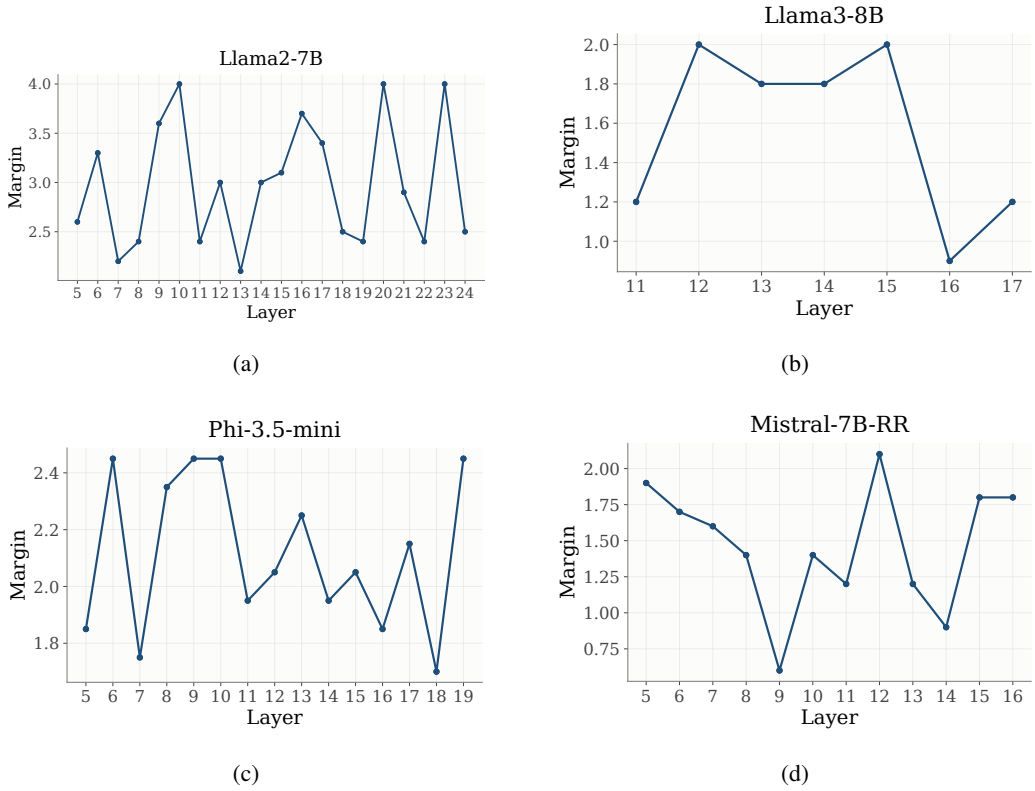


Figure 7: Margin variations across layers

Table 5: BO budget sensitivity across ten models. The retention $\rho_B = \mathrm{ASR}^*(B)/\mathrm{ASR}^*(500)$ measures how much of each model's final best validation ASR was already reached after B trials.

Trials B	**Mean** ρ_B **(%)**	**Median (%)**	**Worst (%)**
50	93.6	95.0	75.9
100	94.8	95.0	75.9
150	97.0	97.4	89.7
200	98.8	100.0	94.9
250	99.3	100.0	97.4
300	99.5	100.0	97.4
350	99.5	100.0	97.4
400	99.5	100.0	97.4
450	99.8	100.0	97.5
500	100.0	100.0	100.0

D.2 Sensitivity of Bayesian Optimization

The layer window and margin selection introduces an additional optimization stage whose outcome may depend on the search budget, the particular intervention window, and the validation judge. We examine these aspects by analyzing the BO trials and using the evaluation protocol described above.

Search Budget. We analyze the BO process for ten models and compute the best validation ASR attained after the first B trials, normalized by the best value after 500 trials: $\rho_B = \mathrm{ASR}^*(B)/\mathrm{ASR}^*(500)$. This measures how much of the final observed search objective is already attained at smaller budgets, rather than performance under independently repeated searches. As shown in Tab. 5, the mean retained ASR is 93.6% after 50 trials and 98.8% after 200 trials. At 200

Table 6: Layer windows and margins selected by Bayesian optimization.

Model	# Layers	Layer window	CLE-A m	CLE-P m
LLaMA2-7B	32	5–25	3.1	1.6
LLaMA3-8B	32	11–18	1.5	1.3
Mistral-7B-RR	32	5–17	1.4	0.6
LLaMA3.2-3B	28	11–23	1.2	1.4
Mistral-7Bv0.3	32	12–15	1.2	1.2
Phi3.5-mini	32	5–20	2.2	2.1
Olmo3-7B	32	11–29	1.6	0.8
Qwen2.5-32B	64	3–47	3.0	2.0
Mixtral-8x7B	32	10–28	1.1	0.6
GPT-OSS-20B	24	9–17	9.8	9.8
DeepSeek-R1-8B	32	13–20	1.8	1.4
Qwen3.5-9B	32	12–18	6.2	5.5
Phi-4-15B	40	9–30	2.6	2.0
Gemma3-12B	48	18–29	1.0	1.0
Ministral3-14B	40	13–33	2.0	0.8

trials, the median retention is 100%, while the lowest retention across models is 94.9%. Thus, most of the best observed validation ASR is typically recovered well before the full 500-trial budget.

Layer-window Sensitivity. In ten models, multiple distinct layer windows attain the same highest validation ASR. These optimal windows have 96.0% mean layer overlap and 100% median overlap. Their similarity indicates that the observed best ASR is not tied to a unique choice of window boundaries, although it does not by itself establish stability across independent BO runs.

D.3 Optimized Layer Windows and Margins

Tab. 6 reports the layer windows and the fixed margins selected by Bayesian Optimization. The selected windows are generally concentrated in intermediate layers, while the earliest and latest layers are usually excluded. This is consistent with the intuition that early layers are less directly tied to refusal behavior, while later layers can often inherit the effect of earlier interventions without requiring direct steering. In early exploratory experiments, we observed that manually selecting an intermediate layer window and tuning a shared margin could already achieve high ASR. This suggested that the precise window boundaries might be less critical than the strength of the intervention. Indeed, steering effects propagate across subsequent layers, allowing margin optimization to partially compensate for differences in the selected window. This helps explain why nearby intervention windows can achieve similar ASR after margin optimization, as also suggested by the sensitivity analysis in Sect. D.2. We therefore use a shared intervention window for CLE-P and CLE-A to facilitate comparison, while optimizing their margins independently. Allowing each variant to optimize its own window does not substantially change the resulting ASR. The selected margins are model-dependent, confirming that a fixed confidence level is not sufficient across architectures. Most models require moderate margins, while reasoning or multimodal models such as GPT-OSS-20B and Qwen3.5-9B require larger margins to obtain reliable evasion. We report examples of layer-wise margins in Fig. 7.

E Margin and Confidence

We show here that the margin m_l introduced in Eq. 8 admits a direct interpretation as the classification confidence of the linear probe under a logistic model. Consider a linear probe $\phi(x) = w^\top x + b$ with $\|w\| = 1$. Under the standard logistic link, the predicted probability of refusal is

$$\mathbb{P}(y = 1 \mid x) \propto \sigma(\phi(x)) = \frac{1}{1 + e^{-\phi(x)}}, \tag{10}$$

where σ denotes the sigmoid function. The minimum-confidence update of Eq. 7, which projects x onto the decision boundary $\mathcal{H}$, yields a steered activation $\tilde{x} = x - \phi(x)\, w$ satisfying $\phi(\tilde{x}) = 0$, and hence $P(y = 1 \mid \tilde{x}) = 1/2$: the point of maximum uncertainty.

To obtain a steered activation with a prescribed compliance confidence $\gamma \in (0, 1)$, we require $P(y = 1 \mid \tilde{x}) = 1 - \gamma$, i.e., the probe predicts refusal with probability $1 - \gamma$ (equivalently, compliance with probability γ). Writing $\tilde{x} = x - \alpha(x)\, w$ and imposing this constraint gives

$$\frac{1}{1 + e^{-\phi(x - \alpha(x)\, w)}} = 1 - \gamma. \tag{11}$$

Since ϕ is linear with $\|w\| = 1$, we have $\phi(x - \alpha(x)\, w) = \phi(x) - \alpha(x)$, from which

$$\phi(x) - \alpha(x) = \log\left(\frac{1 - \gamma}{\gamma}\right) = -\log\left(\frac{\gamma}{1 - \gamma}\right), \tag{12}$$

yielding the step size

$$\alpha(x) = \phi(x) + \log\left(\frac{\gamma}{1 - \gamma}\right). \tag{13}$$

Comparing with our parameterization $\alpha_l(\boldsymbol{t}) = \lambda_l(\phi_l(f^{(l)}(\tilde{\boldsymbol{h}}_l)) + m_l)$ from Eq. 8, with $\lambda_l = 1$, we see that the margin m_l corresponds exactly to the log-odds of the target compliance confidence:

$$m_l = \log\left(\frac{\gamma}{1 - \gamma}\right). \tag{14}$$

Setting $m_l = 0$ recovers $\gamma = 1/2$, i.e., the boundary projection of [5] and the DeepFool attack [9] on linear classifiers. Positive margins push the steered activation deeper into the compliant half-space with increasing confidence: for instance, $m_l \approx 2.94$ corresponds to $\gamma = 0.95$, while $m_l \approx 4.60$ yields $\gamma = 0.99$. This provides a principled interpretation of the margin search in Eq. 9: optimizing m_l via Bayesian Optimization is equivalent to selecting, per layer, the probe compliance confidence that maximizes attack success.

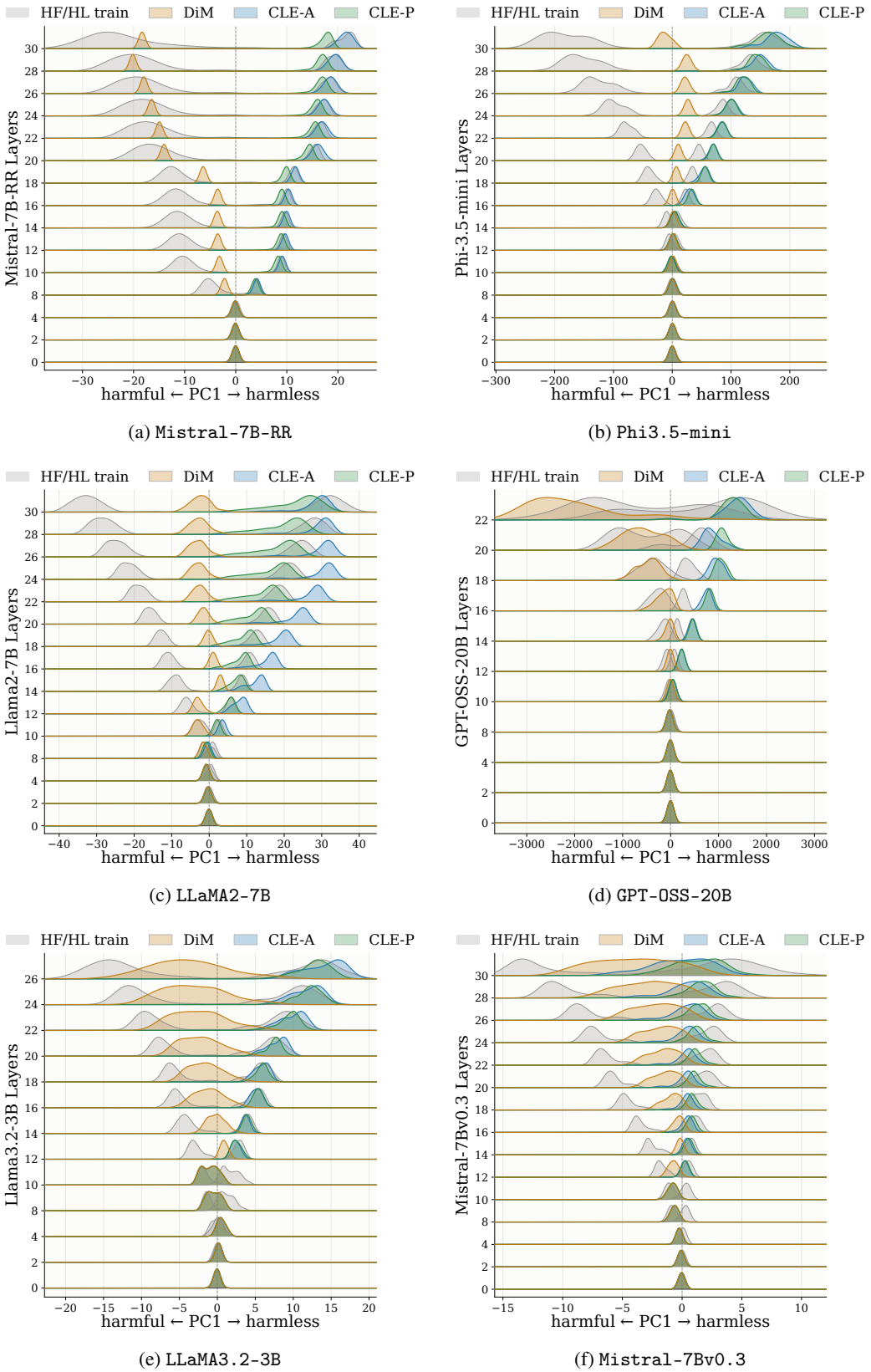
F Additional Mechanistic Analysis

In this section, we provide additional details on the mechanistic analysis reported in Sect. 4. We focus on two complementary views of the latent-space effect induced by CLE. First, we inspect how the intervention moves the post-instruction representation across layers. Second, we analyze whether the induced shift is preserved during autoregressive generation.

Layer-wise prompt representations. Fig. 8 reports ridgeline plots of post-instruction activations across layers. For each model and layer, we collect the last-token residual-stream activations of harmful and harmless prompts used to train the probes, together with the activations obtained on HARMBENCH test prompts under DiM ablation, CLE-P, and CLE-A. We then project the activations of each layer onto the first principal component, computed from the reference harmful/harmless activations at that layer. The sign of the component is chosen so that harmless representations lie on the positive side of the axis. Each ridge shows, for a fixed layer, where the harmful test prompts are placed relative to the harmful (refusal) and harmless (compliance) training distributions.

These plots provide a direct latent-space view of the difference between minimum-confidence ablation and controlled evasion. DiM often leaves the harmful test distribution close to its original refusal region, or only weakly moves it toward the decision boundary. This is consistent with our formulation: DiM ablation corresponds to a boundary projection and does not explicitly optimize for a positive compliance margin. In contrast, both CLE-P and CLE-A shift the same prompts toward the harmless side of the representation space, often making their distribution overlap with the harmless training prompts. This confirms that the optimized margins do not merely remove refusal evidence, but push representations into the compliant region. The effect is especially visible in intermediate and late layers, where harmful and harmless representations are more separated. Early layers are instead often less informative, with overlapping harmful/harmless distributions, which explains why the BO search tends not to select them.

Generated-token trajectories. Fig. 9 complements the layer-wise prompt analysis by tracking the latent trajectory during generation. For each harmful prompt of the HARMBENCH dataset, we generate completions under DiM, CLE-P, and CLE-A. At each generated token position, we record the last-layer activation and evaluate it with the corresponding linear probe. We convert the

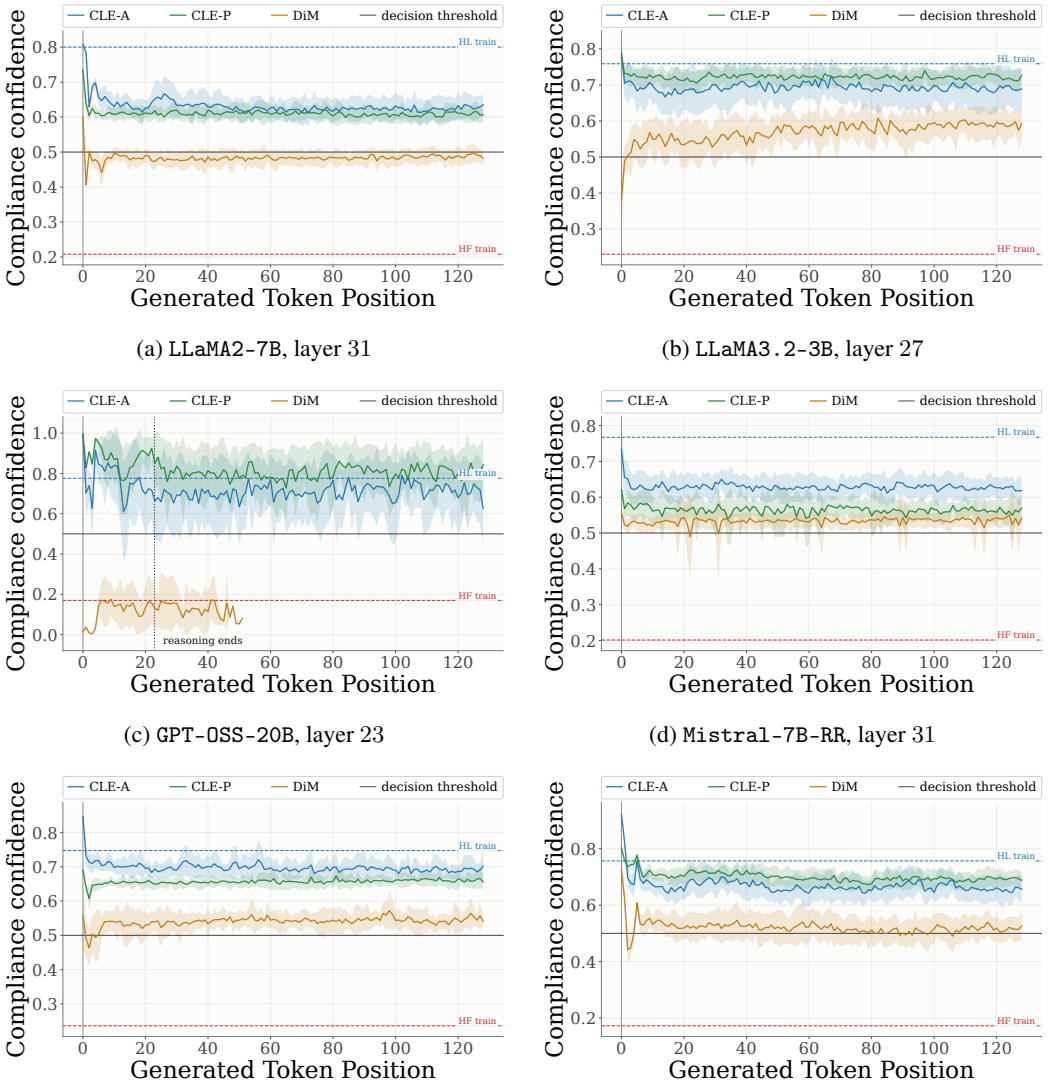


(a) `Mistral-7B-RR` (b) `Phi3.5-mini`

(c) `LLaMA2-7B` (d) `GPT-OSS-20B`

(e) `LLaMA3.2-3B` (f) `Mistral-7Bv0.3`

Figure 8: 1st PC of prompt activation across layers: CLE variants confidently shift test prompts into the harmless, compliant region, while DiM leaves activation distribution nearly unchanged.



(a) `LLaMA2-7B`, layer 31

(b) `LLaMA3.2-3B`, layer 27

(c) `GPT-OSS-20B`, layer 23

(d) `Mistral-7B-RR`, layer 31

(e) `Olmo3-7B`, layer 30

(f) `DeepSeek-R1-8B`, layer 23

Figure 9: Generated tokens from HARMBENCH prompts classified by last layer probes

probe score into compliance confidence as $1 - \sigma(\phi_l(h))$, so that values above the decision threshold indicate movement toward the compliant side of the probe. The curves report the average compliance confidence across prompts at each generated token position, while the horizontal reference lines indicate the average confidence of harmful and harmless probe-training prompts.

The token-level plots show that CLE-A preserves a compliant trajectory throughout generation, despite applying only a fixed perturbation computed at the post-instruction token. This is important because CLE-A does not reproject each newly generated activation onto the shifted probe boundary. Nevertheless, its generated-token activations remain consistently above the decision threshold and close to the harmless reference region. In several models, CLE-A matches or exceeds CLE-P, suggesting that continuous reprojection is not required once the initial latent trajectory has been moved into the compliant region.

By contrast, DiM generally remains close to the harmful region or oscillates around low compliance-confidence values. This supports the interpretation that minimum-confidence ablation is too weak: it may reduce local refusal evidence, but it does not reliably induce a stable, compliant trajectory during decoding. Overall, the ridgeline and token-generation analyses support the same conclusion from

two perspectives: controlled evasion changes where the prompt representation lies in latent space, and this shift is then propagated through generation, especially for the additive CLE-A variant.

G Additional Ablation Analysis

Fig. 10 and Fig. 11 ablate the main components of CLE-P and CLE-A. Starting from the full method introduced in Sect. 3, we remove one design choice at a time: we replace the SVM probes with DiM probes, remove layer selection by setting $\lambda_l \equiv 1$ (i.e. intervening on all layers), remove confidence control by setting $m_l \equiv 0$, and finally remove both layer selection and confidence control by setting $\lambda_l \equiv 1$ and $m_l \equiv 0$. This allows us to test whether the performance gain comes from the probe quality, the choice of intervention layers, the positive evasion margin, or their combination.

The results show the same trend for both CLE-P and CLE-A. Using DiM instead of SVM probes reduces ASR, confirming that a more accurate latent surrogate gives a stronger evasion direction. Setting $\lambda_l \equiv 1$ is also suboptimal: even with optimized margins, intervening on all layers can perturb layers where refusal information is weak or where steering compromises the generation. Setting $m_l \equiv 0$ is weaker as well, because it results in projecting activations onto the probe boundary, recovering a minimum-confidence evasion attack rather than pushing them into the compliant region. The weakest variant is generally the one with both $\lambda_l \equiv 1$ and $m_l \equiv 0$, which corresponds to all-layer boundary projection.

Overall, these ablations show that the gains of CLE-P and CLE-A are not explained by a single component. Strong refusal suppression requires the combination of accurate probes, selective layer intervention, and positive margin control.

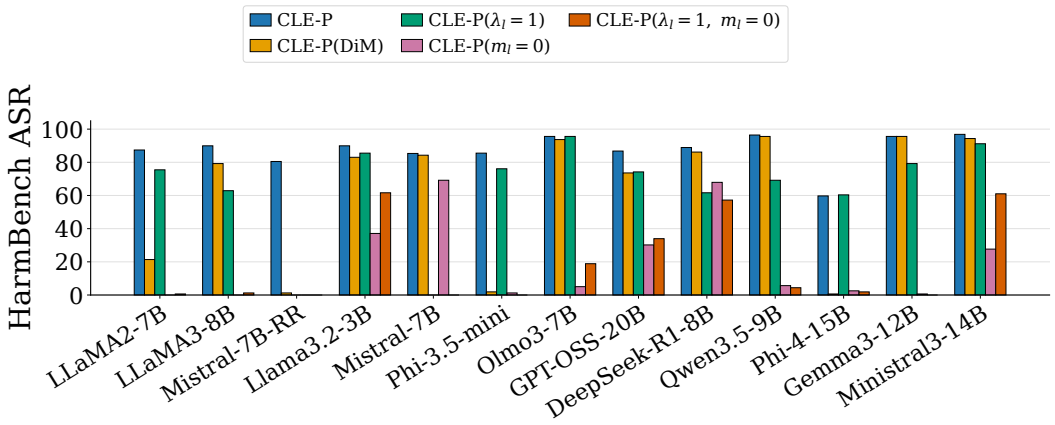


Figure 10: Ablation study among CLE-P components.

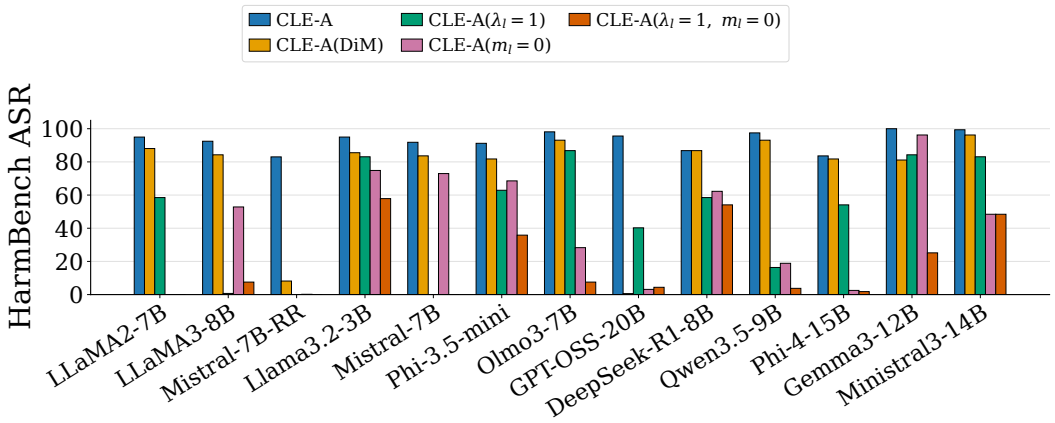


Figure 11: Ablation study among CLE-A components.

Table 7: Matched-margin experiment (ASR, %): CLE-P is evaluated with its own margin and with the margin optimized for CLE-A. Results are shown for ten selected models with different optimized margins.

Model	**CLE-A**	**CLE-P**	**CLE-P at CLE-A margin**
`LLaMA2-7B`	94.96	87.42	28.30
`LLaMA3-8B`	92.45	89.94	86.79
`Mistral-7B-RR`	83.02	80.50	9.43
`LLaMA3.2-3B`	94.97	89.94	88.05
`Phi3.5-mini`	91.20	85.53	81.13
`Olmo3-7B`	98.11	95.60	52.20
`DeepSeek-R1-8B`	86.79	84.91	77.36
`Qwen3.5-9B`	97.48	92.45	89.31
`Phi-4-15B`	78.61	75.47	45.91
`Ministral3-14B`	99.37	96.85	83.02
Average	91.70	87.86	64.15

G.1 Comparing Projection and Addition at Matched Margins

We next compare the margin choice between the two CLE methods. Although CLE-A generally achieves comparable or higher ASR than CLE-P, the two variants can select different evasion margins on the same layer window. In particular, CLE-A often uses a larger margin, raising the possibility that its advantage comes from stronger steering rather than from the additive intervention itself. To test this explanation, we reevaluate CLE-P using the margins optimized for CLE-A, while keeping the intervention window fixed. The comparison covers ten models with distinct optimized margins among the 13 models considered in this analysis. As reported in Tab. 7, replacing CLE-P's optimized margins with those of CLE-A reduces its mean ASR from 87.86% to 64.15%, whereas CLE-A obtains 91.70%. Hence, matching the margin does not close the gap between projection and addition; instead, the two interventions appear to operate best at different margin regimes. One possible explanation follows from their effects during decoding: CLE-P repeatedly projects each token activation to the shifted probe boundary, whereas CLE-A applies a fixed prompt-dependent perturbation that need not place every subsequent activation on that boundary. A margin that is beneficial for the latter can therefore over-constrain the former. The associated effect on model capability is substantial in some cases. On `LLaMA2-7B`, increasing the CLE-P margin from 1.6 to 3.1 reduces ASR from 87.42% to 28.30%, while ARC accuracy drops from 58.16% to 28.84% (61.51% without intervention) and MMLU accuracy from 46.74% to 37.31%. This joint decline in ASR and task accuracy is consistent with excessive steering disrupting generation, rather than simply making refusal suppression stronger.

Table 8: Model accuracy on MMLU, ARC, and TruthfulQA benchmarks before (gray) and after CLE-P intervention.

Model	MMLU	ARC	TRUTHFULQA
`LLaMA2-7B`	46.74 47.24	58.16 61.51	27.29 30.11
`LLaMA3-8B`	65.10 67.14	79.69 82.25	34.23 37.08
`Mistral-7B-RR`	53.30 56.88	64.84 76.96	48.86 53.36
`LLaMA3.2-3B`	60.49 61.20	75.08 75.00	29.62 33.53
`Mistral-7Bv0.3`	58.87 62.07	74.91 78.66	36.45 41.98
`Phi3.5-mini`	62.71 68.96	81.14 87.96	29.53 38.43
`Olmo3-7B`	54.69 55.12	36.46 40.27	29.88 38.43
`Qwen2.5-32B`	79.24 83.21	68.69 73.63	58.45 65.55
`Mixtral-8x7B`	69.47 69.25	63.20 62.00	61.90 64.09
`GPT-OSS-20B`	73.19 84.59	76.26 74.81	64.45 72.98
`DeepSeek-R1-8B`	56.44 56.41	47.18 49.40	45.86 50.88
`Qwen3.5-9B`	70.46 80.38	64.86 70.82	53.58 63.68
`Phi-4-15B`	70.76 77.66	54.78 65.87	50.38 55.51
`Gemma3-12B`	69.67 72.17	69.28 71.42	49.79 57.97
`Ministral3-14B`	77.87 78.79	68.94 71.59	51.94 58.83

Table 9: Model accuracy on MMLU, ARC, and TruthfulQA benchmarks before (gray) and after CLE-A intervention.

Model	MMLU	ARC	TRUTHFULQA
`LLaMA2-7B`	43.71 47.24	53.54 61.51	26.88 30.11
`LLaMA3-8B`	64.90 67.14	79.43 82.25	33.21 37.08
`Mistral-7B-RR`	47.04 56.88	54.18 76.96	42.82 53.36
`LLaMA3.2-3B`	61.03 61.20	74.82 75.00	32.06 33.53
`Mistral-7Bv0.3`	58.42 62.07	76.36 78.66	38.00 41.98
`Phi3.5-mini`	65.04 68.96	82.14 87.96	33.53 38.43
`Olmo3-7B`	52.52 55.12	31.03 40.27	27.03 38.43
`Qwen2.5-32B`	76.06 83.21	62.88 73.63	47.36 65.55
`Mixtral-8x7B`	69.42 69.25	61.58 62.00	58.69 64.09
`GPT-OSS-20B`	72.49 84.59	76.01 74.81	64.48 72.98
`DeepSeek-R1-8B`	56.25 56.41	46.50 49.40	45.02 50.88
`Qwen3.5-9B`	69.46 80.38	59.88 70.82	53.48 63.68
`Phi-4-15B`	65.50 77.66	53.59 65.87	48.41 55.51
`Gemma3-12B`	71.42 72.17	70.31 71.42	49.46 57.97
`Ministral3-14B`	74.88 78.79	63.80 71.59	44.63 58.83

H Model Capabilities and Output Coherence

Refusal suppression is useful as an evaluation of safety mechanisms only if high ASR does not arise primarily from severely degraded generation. Since CLE intervenes directly in the residual stream, we assess its effect on general model capabilities using MMLU, ARC, and TruthfulQA under the same inference-time steering used for the attack experiments. The following results cover all 15 models and compare the unsteered models with CLE-P and CLE-A.

Tab. 8 and Tab. 9 show that many models retain substantial task accuracy, but the cost of steering varies across models and benchmarks. In particular, on `Mistral-7B-RR`, ARC accuracy decreases from 76.96% without steering to 64.84% under CLE-P and 54.18% under CLE-A. Since this model is trained with Representation Rerouting, its representations may be especially sensitive to interventions, although these measurements alone cannot establish the cause. These benchmarks evaluate *always-on* steering, including on ordinary tasks. Unlike a weight update, CLE can be disabled at inference time, leaving the original model computations unchanged. We therefore interpret the measured capability losses as the cost of activating the intervention, rather than a permanent change to the model. Finally, we examine whether judge-labeled attack successes on `Mistral-7B-RR` might instead be artifacts of incoherent or degenerate completions. We manually inspect all responses classified as successful under CLE-A and find no case in which the success label can be explained solely by degeneration

or by the absence of an explicit refusal. The inspected responses are coherent and relevant to the requested behavior.

I Computing

For models up to 15B parameters, we ran all experiments on a single NVIDIA RTX A6000 GPU with 48GB of memory. For larger models, we used a single NVIDIA H100 GPU with 94GB of memory. On average, the Bayesian Optimization search required approximately 6 hours on one GPU when evaluated on the HARMBENCH validation set. The remaining steps of the pipeline were substantially cheaper. Training the linear probes, instantiating the intervention, and generating completions with the steered model required only a few minutes beyond standard generation time. In particular, once the optimized layer window and margin were selected, inference with CLE-P and CLE-A ran at the model's usual generation cost, since the intervention only modifies activations during the forward pass and does not require additional optimization at test time.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims of the paper are supported by theoretical and empirical evidence in Sect. 2, Sect. 3, Sect. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Sect. 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We report complete proofs in Sect. 2 and App. E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all experimental settings in Sect. 4 and App. B, along with the codebase in `https://github.com/pralab/latent-evasion`, to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide all experimental settings in Sect. 4 and App. B to reproduce experimental results. Our code has been publicly released at `https://github.com/pralab/latent-evasion` and, prior to acceptance, anonymized in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all experimental settings in Sect. 4 and App. B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main results of the paper, reported in Sect. 4, are computed over multiple random seeds and include the corresponding standard deviations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report information on computer resources in App. I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We discuss the ethical impact of our work in App. A.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impact of our work in App. A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We address responsible release App. A.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the authors and original owners of all methods, datasets, and models used in our work, and we explicitly respect their corresponding licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the codebase developed for this work and provide documentation alongside it, including instructions for setup, usage, and reproduction of the experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Although the paper is centered on testing LLMs under several scenarios, they have not been used within the core methodology, and they do not impact scientific rigor or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.