CONCEPT CONTROL IN LANGUAGE MODELS VIA ACTIVATION STRAIGHTENING AND CLAMPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation intervention aims to control concepts in LLMs by intervening on internal activations. While promising, existing methods often compromise general model utility or fail to generalize under distribution shifts. We argue that these failures of faithful control stem from two structural obstacles in latent space. First, concepts are redundantly distributed across layers. Sparse interventions allow the model to bypass the intervention (the "hydra effect"), whereas naively intervening at many layers can over-correct activations and degrade utility. Second, the activation patterns that encode a concept can vary with context, so a simple linear subspace often fails to capture the geometry of the concept and can perturb unrelated capabilities, especially on out-of-distribution prompts. We propose Straighten-and-Clamp (`SAC`). `SAC` learns an invertible straightening map, which concentrates concept-related variation, followed by a projection clamp to enforce a target concept code in the straightened coordinates. Because the clamp is idempotent at each site, it becomes inactive once the target code is already satisfied, enabling stable multi-layer deployment. We train `SAC` in two stages: first, we learn a straightening map amortized across concepts; second, we learn lightweight per-concept clamps via pairwise preference optimization. Experiments on concept induction and suppression of harmful concepts suggest that `SAC` improves the control–utility trade-off and robustness to distribution shifts.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed in complex environments (Grattafiori et al., 2024; Guo et al., 2025; Yang et al., 2025a; Cao et al., 2026), where developers require targeted concept control over model responses. The goal is often twofold: suppressing harmful concepts (e.g., illegal activity, hate speech) to ensure safety (Han et al., 2024; Chao et al., 2024; Xie et al., 2025), or inducing benign concepts (e.g., a specific writing style, persona, or domain framing) to customize generation. Achieving concept-level control is challenging because these concepts are not explicit variables. They are implicit in high-dimensional internal representations that interact intricately with the model's general capabilities (Mueller et al., 2025). This challenge has motivated a line of work on concept-centric alignment and representation-level intervention (Geiger et al., 2023; Wu et al., 2023; Geiger et al., 2024; Wu et al., 2024), which seeks to identify concept-relevant representations and intervene directly on these activations to steer or align model behavior.

A growing family of representation intervention methods has emerged to address this, utilizing techniques ranging from static steering vectors or refusal directions (Rimsky et al., 2024; Arditi et al., 2024), dictionary-based concept engineering (Luo et al., 2024), and rotation-based geometric steering (Vu and Nguyen, 2025) to learned affine transformations, such as Representation Fine-Tuning (ReFT) (Wu et al., 2024) and Localized Fine-tuning (LoFiT) (Yin et al., 2024), and distributional activation-steering methods such as LinEAS (Rodriguez et al., 2025). Despite their different parameterizations, these methods typically edit activations in the native representation geometry, using fixed directions, dictionaries, affine maps, or low-dimensional steering planes. Although lightweight and modular compared to full fine-tuning, these methods frequently struggle with faithfulness. A faithful intervention should reliably control the target concept while minimally degrading the general utility. It should also remain robust under a prompt distribution shift. We argue that faithfulness failures in existing representation intervention methods largely stem from two structural obstacles.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The first obstacle is *the redundancy dilemma*. Concepts in modern LLMs are typically distributed across network depth. Intervening at a few locations is brittle because the model can route information through alternative pathways, which is a phenomenon known as the hydra effect (McGrath et al., 2023). A natural response is to intervene densely across layers, but standard additive shifts and unconstrained affine interventions generally do not impose the desired concept-constraint set as their fixed-point set, nor do they guarantee idempotency. Repeated application of such interventions can therefore accumulate unnecessary intervention effects even when a local state is already aligned. This over-correction may move activations outside their typical distribution and degrade utility.

The second obstacle is the *context-dependent structure of the concept geometry*. Most interventions rely on the linear representation hypothesis (Park et al., 2023), assuming that a concept can be controlled by manipulating a fixed linear direction or subspace in activation space. In practice, the effective concept direction can bend across the activation space, consistent with the evidence that semantic concepts may exhibit non-linear geometric structure Ravfogel et al. (2022b); Modell et al. (2025). As a result, a single linear subspace is often only locally valid. A linear intervention learned in one distribution can, therefore, entangle the target concept with other features under distribution shift, leading to weak control and failure.

To address these obstacles, we propose Straighten-and-Clamp (`SAC`), which changes the intervention primitive from "adding a vector" to "clamping a constraint" in a straightened geometry. `SAC` uses a projection-based clamp. At an intervention site, the representation is projected onto the set of states whose concept coordinates match a target concept code. This clamp has a fixed point and is idempotent. In a multi-layer deployment, each clamp computes the current concept coordinates and corrects only the drift that the model has regenerated since the previous intervention site. If the incoming representation already satisfies the target code at that site, the clamp does nothing, which avoids repeated over-correction by the same operator. To make clamping faithful when the concept geometry is curved, `SAC` learns an invertible straightening map to concentrate the concept variations, making the concept geometry more amenable to linear intervention. We perform clamping in this straightened space and map back via exact inverse. A two-stage training strategy is proposed to decouple geometric rectification from specific concept control. Stage 1 learns a straightening map amortized across many concepts, optimizing for subspace concentration. Stage 2 freezes the map and learns lightweight, per-concept parameters to define the clamp targets.

Our contributions can be summarized as follows:

- We propose Straighten-and-Clamp (`SAC`), an intervention framework that combines an invertible straightening map with an idempotent projection clamp. This design enables dense deployment across layers while avoiding repeated over-correction, thereby effectively addressing the redundancy dilemma behind the hydra effect.
- We introduce a two-stage training strategy that separates geometry learning from concept-specific control. The first stage learns a shared straightening map to concentrate concept-related variation, while the second stage learns lightweight per-concept clamps for scalable concept control by using a preference-based objective.
- We validate `SAC` on concept induction and harmful concept suppression benchmarks. Empirical results demonstrate that `SAC` outperforms state-of-the-art baselines in the control–utility trade-off, while exhibiting superior robustness against distribution shifts.

2 Related Work

Causal Interpretability and Concept Alignment. Mechanistic interpretability (Sharkey et al., 2025; Mueller et al., 2025) operates on the premise that neural networks execute high-level algorithms, treating concepts as internal variables within a causal model (Geiger et al., 2023). A central objective in this field is to map these abstract concepts to specific subspaces within the model's representation (Geiger et al., 2024; Wu et al., 2023). While linear probes have traditionally been used to detect semantic information, they often fail to distinguish between features that are causally necessary for generation and those that are merely correlated (Hase et al., 2023). To establish stronger causal links, frameworks such as causal abstraction (Geiger et al., 2023) have been developed to verify the functional role of specific neural components. Furthermore, recent advances in sparse autoencoders (SAEs) (Lieberum et al., 2024; Rajamanoharan et al., 2024; Cunningham et al., 2023)

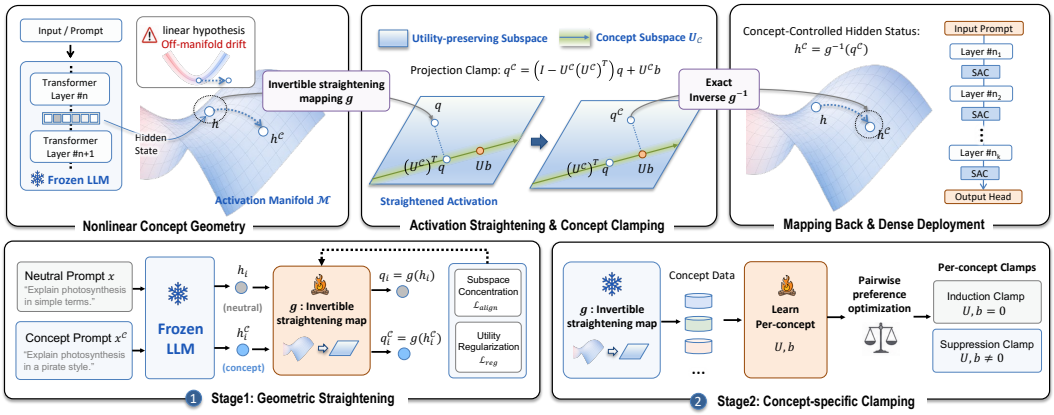


Figure 1: **Overview of the SAC intervention framework.** The method consists of two main stages. **Stage 1** learns an invertible straightening map to transform the non-linear concept geometry from the frozen LLM's hidden states into a straightened activation manifold, optimizing for both concept subspace concentration and utility preservation. **Stage 2** freezes the straightening map and learns lightweight, per-concept projection clamps via pairwise preference optimization. The clamped representations are mapped back to the native space via the exact inverse and deployed densely.

enable the unsupervised discovery of fine-grained feature dictionaries. However, these methodologies are predominantly diagnostic. Our work bridges the gap between understanding and manipulation, leveraging these causal insights to engineer interventions that actively control model behavior.

Representation-level Intervention. Building on interpretability findings, representation intervention (Wu et al., 2024; Yin et al., 2024; Wu et al., 2025b; Rodriguez et al., 2025) aims to localize model behaviors onto representations that encode high-level concepts, enabling control by manipulating these internal states. A common paradigm involves adding steering vectors derived from contrastive activations to the residual stream (Rimsky et al., 2024). Recent advancements have introduced more sophisticated techniques. For instance, linear concept-erasure methods such as INLP (Ravfogel et al., 2020), R-LACE (Ravfogel et al., 2022a), and LEACE (Belrose et al., 2023) remove or neutralize concept subspaces. To improve intervention effectiveness, methods such as ReFT and LoFiT learn affine transformations to induce or suppress concepts. However, these approaches suffer from two structural limitations that create faithfulness gaps. First, because concepts are often redundantly distributed, sparse interventions allow the model to re-route the concept via alternative pathways (McGrath et al., 2023; Wang et al., 2022), whereas dense linear interventions accumulate excessive drift that degrades general utility. Second, these methods rely on the linear representation hypothesis. Recent geometric analyses (Gonen and Goldberg, 2019; Ravfogel et al., 2022b; Modell et al., 2025) suggest that semantic concepts often reside on non-linear manifolds, meaning that the affine transformations used by current methods cannot perfectly isolate the target concept without distorting the natural geometry of the data. More recent work, such as COCA (Yang et al., 2025b), argues that non-linear entanglement can prevent faithful concept erasure and mitigate this issue at the data level via improving how concepts are organized during training. This data-level strategy is complementary to SAC, which addresses the intervention-level geometry and redundancy dilemma.

3 Intervention Method

3.1 Problem Setup

We consider a frozen autoregressive language model with distribution $p_0(y \mid x)$, where x is the prompt and y is the model response. Fix a layer l and token position t, we write $s := (l, t)$ for the corresponding intervention site. Let $h_s \in \mathbb{R}^d$ denote the hidden state at this site, such as the residual-stream representation, produced when the model processes x. For any replacement vector $z \in \mathbb{R}^d$, we denote by $p_0(\cdot \mid x; h_s \leftarrow z)$ the output distribution obtained by running the frozen model on x while replacing h_s with z. A *representation intervention* is a function $f : \mathbb{R}^d \to \mathbb{R}^d$ that edits the hidden state from h_s to $f(h_s)$, thereby inducing the intervened distribution

$p_{\text{int}}(\cdot \mid x) := p_0\big(\cdot \mid x;\, h_s \leftarrow f(h_s)\big)$. The same notation extends naturally to interventions at multiple sites. For simplicity, we omit the site index s in the following discussion.

Concept-centric alignment asks whether we can control a model's behavior with respect to a high-level concept by editing its internal representations. For a fixed model and intervention site, we assume that the concept-related information is decodable from the hidden state through a concept readout $\varphi : \mathbb{R}^d \to \mathbb{R}^r$ with $r \ll d$, producing a concept code $c = \varphi(h)$. The goal of representation intervention is to learn an intervention f such that the intervened distribution $p_{\text{int}}(\cdot \mid x)$ generates responses satisfying a desired concept property, such as inducing or suppressing the concept. We call an intervention *faithful* if it reliably achieves the target concept property while preserving general utility and remaining robust under shifts in the prompt distribution.

3.2 Design Motivation and Principles

Principle 1: The intervention operator should have fixed points. To faithfully control a concept, dense deployment is often useful against the hydra effect, but it is safe only if the intervention leaves already aligned states unchanged. In particular, the intervention should become a no-op once the incoming state already matches the target concept code. We formalize this with a per-site *target constraint set* $\mathcal{H} \subseteq \mathbb{R}^d$ of hidden states that satisfy the desired concept constraint. We want the intervention f to behave like a projection onto the target constraint set. That is, it should map every hidden state into $\mathcal{H}$, while leaving states already in $\mathcal{H}$ unchanged, i.e.,

$$\forall h \in \mathbb{R}^d,\ f(h) \in \mathcal{H}, \quad \text{and} \quad \forall h \in \mathcal{H},\ f(h) = h. \tag{1}$$

These two conditions mean that $\mathcal{H}$ is exactly the fixed-point set of f, and they imply per-site *idempotence*, that is, $f(f(h)) = f(h),\ \forall h \in \mathbb{R}^d$. If the state already lies in $\mathcal{H}$, the intervention is a no-op. Consequently, it only corrects the remaining concept-coordinate drift at that site, and avoids further changing an already aligned representation (see Proposition 3.1).

A simple way to satisfy the requirement equation 1 is to define the intervention as a projection onto the desired concept-constraint set. Following (Wu et al., 2024; Park et al., 2023) as a motivating case, assume that the concept code admits a local linear readout $c = \varphi(h) = U^\top h$, where $U \in \mathbb{R}^{d \times r}$ has orthonormal columns (i.e., $U^\top U = I_r$). Given a desired concept code $b \in \mathbb{R}^r$, we define the clamp as the projection of h onto the constraint set $\mathcal{H}_{(U,b)} := \{h' \in \mathbb{R}^d : U^\top h' = b\}$, which admits a closed-form solution:

$$\Phi_{(U,b)}(h) = \operatorname*{arg\,min}_{h' \in \mathcal{H}_{(U,b)}} \|h' - h\|_2^2 = (I - UU^\top)h + Ub. \tag{2}$$

Since $\Phi_{(U,b)}$ is an orthogonal projection, it is idempotent and leaves any $h' \in \mathcal{H}_{(U,b)}$ unchanged.

Principle 2: straightening before clamping. Idempotent clamping is most faithful when the concept geometry is close to a linear subspace. As discussed in Section 1, in the native activation space, the effective concept direction can bend with context, so a single linear subspace is often only locally valid. Recall that in Section 3.1, the relationship between a hidden state $h \in \mathbb{R}^d$ and a concept code $c \in \mathbb{R}^r$ is modeled by a concept readout $\varphi : \mathbb{R}^d \to \mathbb{R}^r$, where $c = \varphi(h)$. If the concept readout $\varphi : \mathbb{R}^d \to \mathbb{R}^r$ is continuously differentiable and has a locally constant rank, then by the rank Theorem (Spivak, 2018, Theorem 2-13), there exist local diffeomorphisms that transform the concept readout into a coordinate projection. In other words, after a suitable local change of variables, the concept can be isolated into a fixed set of r coordinates. In that idealized geometry, concept control is as simple as clamping those coordinates to a desired target while leaving the rest untouched.

This motivates us to learn an invertible straightening map $g : \mathbb{R}^d \to \mathbb{R}^d$ that approximately performs this coordinate simplification (see Appendix E.1 for the formal statement). Invertibility is essential. We apply the clamp in the straightened $\{g(h)\}_h$-space and map back via $f(h) = g^{-1}\big(\Phi_{(U,b)}(g(h))\big)$. The clamping preserves the component of $g(h)$ orthogonal to the concept subspace exactly. After mapping back through g^{-1}, the native coordinates can change, but the update remains structured: the clamp is applied in an invertible coordinate system rather than through an arbitrary overwrite.

In the straightened space $\{g(h)\}_h$, the concept variation is encouraged to concentrate in a linear subspace. We do not assume that a single global diffeomorphism exactly straightens φ everywhere. Even when no single global diffeomorphism perfectly straightens the concept, the learned map can still serve as a preconditioner that reduces local curvature and makes linear intervention more faithful.

In the following Proposition 3.1, the overall intervention f leaves the already aligned states unchanged and corrects only the newly regenerated drift. The proof can be found in Appendix F.

Proposition 3.1 (Constraint satisfaction, fixed points, idempotence, and drift removal)**.** *Assume $g : \mathbb{R}^d \to \mathbb{R}^d$ is invertible and $U \in \mathbb{R}^{d\times r}$ has orthonormal columns, i.e., $U^\top U = I_r$. For a desired concept code $b \in \mathbb{R}^r$, define the affine clamp in the straightened coordinates by $\Phi_{(U,b)}(q) := (I - UU^\top)q + Ub,\ q \in \mathbb{R}^d$, and define the corresponding intervention $f(h) := g^{-1}\big(\Phi_{(U,b)}(g(h))\big)$. Let $\mathcal{H}_{g,U,b} := \{h \in \mathbb{R}^d : U^\top g(h) = b\}$ be the target concept-constraint set in the native coordinates. Then, for any $h \in \mathbb{R}^d$, letting $q := g(h)$ and $\delta := U^\top q - b$, the following properties hold:*

(i) ***Constraint satisfaction.*** *The intervened straightened representation satisfies $U^\top \Phi_{(U,b)}(q) = U^\top g(f(h)) = b$, and hence it implies that $f(h) \in \mathcal{H}_{g,U,b}$.*

(ii) ***Fixed points and idempotence.*** *The fixed-point set of f is exactly $\mathcal{H}_{g,U,b}$, that is, $f(h) = h$ if and only if $U^\top g(h) = b$. Moreover, $\Phi_{(U,b)}(\Phi_{(U,b)}(q)) = \Phi_{(U,b)}(q)$, $f(f(h)) = f(h)$.*

(iii) ***Drift removal.*** *The straightened representation after intervention is $g(f(h)) = g(h) - U\delta$. Equivalently, the clamp removes the concept-coordinate residual $\delta = U^\top g(h) - b$ while preserving the orthogonal component: $(I - UU^\top)g(f(h)) = (I - UU^\top)g(h)$.*

3.3 Intervention Design

Orthogonal Clamp. At each intervention site, we apply the clamp $\Phi_{(U,b)}(g(h)) = (I - UU^\top)g(h) + Ub$ in the straightened space, and map back with g^{-1}. We constrain U to have orthonormal columns so that $\Phi_{(U,b)}$ is idempotent. To optimize U while keeping it orthonormal, we re-parameterize it using a proxy parameter matrix $Z \in \mathbb{R}^{d\times r}$. This allows us to optimize Z with standard gradient methods, and then compute a corresponding U that satisfies $U^\top U = I_r$ without explicit manifold retractions. Specifically, let $G = Z^\top Z$ and let $\rho = \|G\|_F$. We scale Z by ρ, setting $\widetilde{Z} = Z/\sqrt{\rho}$ and $\widehat{G} = \widetilde{Z}^\top \widetilde{Z}$. Starting from $R_0 = I_r$, we compute $R_{k+1} = \frac{1}{2} R_k\big(3I_r - \widehat{G}R_k^2\big)$, where $k = 0, 1, \ldots, K-1$. We then set $U = \widetilde{Z}R_K$, which gives $U^\top U \approx I_r$ (see (Higham, 1986; Huang et al., 2020)) with a modest K.

Straightening Map. To operationalize Principle 2, we parameterize g as an exactly invertible normalizing flow, instantiated either as NICE (Dinh et al., 2014) or as a non-linear Householder flow (HF). Both variants admit exact inverses and yield stable numerical conditioning in practice. The non-linear Householder flow can achieve runtime comparable to ReFT. However, the rotation matrix transformation implemented in ReFT cannot, in general, align a context-dependent concept geometry into a linear subspace. Formal statements, derivations and architectural details are in Appendix E.

Layer-wise SAC modules. We attach `SAC` to a set of intervened layers $\mathcal{L}_{\text{int}} \subseteq \{1, \ldots, L\}$. Each layer $l \in \mathcal{L}_{\text{int}}$ has its own invertible straightening map g_l, concept subspace basis U_l (with orthonormal columns $U_l^\top U_l = I_r$), and the target concept code b_l. When the discussion concerns a single intervened layer, we omit the layer subscript and write g, U, b, f instead of g_l, U_l, b_l, f_l. We reinstate layer indices only when comparing or composing multiple layers and when defining objectives that aggregate across layers $l \in \mathcal{L}_{\text{int}}$.

4 Two-Stage Training for Concept-Specific Intervention

Data Format. Let i denote the data sample index and $\mathcal{C}$ denote a concept. For concept induction (turning a concept on), we use a dataset $\mathcal{D}_{\text{ind}} = \{(x_i^{\mathcal{C}}, x_i, y_i^{\mathcal{C}}, y_i)\}_{i=1}^{N_{\mathcal{C}}}$. Here, x_i is a neutral instruction, y_i is a neutral response, $x_i^{\mathcal{C}}$ is an instruction designed to trigger concept $\mathcal{C}$, and $y_i^{\mathcal{C}}$ is a response that induces concept $\mathcal{C}$. For concept suppression (turning a concept off), we use $\mathcal{D}_{\text{supp}} = \{(x_i^{\mathcal{C}}, x_i, y_i^{\mathcal{C}}, y_i^{\neg\mathcal{C}})\}_{i=1}^{N_{\mathcal{C}}}$, where $y_i^{\neg\mathcal{C}}$ is a response where concept $\mathcal{C}$ is explicitly suppressed. Additionally, we utilize a generic instruction-tuning dataset $\mathcal{D}_{\text{sft}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{sft}}}$ for utility preservation. We include data examples in Appendix G.2.

Intervention Settings. We define two distinct control settings based on the desired concept code b. These settings determine the intervened model p_{int}. **Concept induction** aims to actively trigger or amplify a concept. By setting $b \neq 0$, we force the representation's projection onto $\text{span}(U)$ to be exactly Ub (equivalently, $U^\top h' = b$, as in Eq. equation 2). **Concept Suppression** aims to suppress a concept from the representation and we set $b = 0$.[1] We denote the intervened model under the induction setting as p_{ind} and under the suppression setting as p_{supp}. We work with response length normalized log-likelihood scores $s_{\text{ind}}(x, y) = \frac{1}{|y|} \log p_{\text{ind}}(y|x)$ and $s_{\text{supp}}(x, y) = \frac{1}{|y|} \log p_{\text{supp}}(y|x)$.

Two-stage Training. We adopt a two-stage training strategy to decouple population-level geometric rectification from concept-specific intervention. We attach SAC to a fixed set of intervened layers $\mathcal{L}_{\text{int}}$. Stage 1 learns the layer-wise population-level straightening map g_l, facilitating faithful control of the concept using a linear clamp. Stage 2 then learns the layer-wise concept-specific clamp parameters $(U_l^{\mathcal{C}}, b_l^{\mathcal{C}})$ for a concept $\mathcal{C}$, while keeping the straightening map fixed. When the discussion concerns a single fixed layer l, we drop the layer subscript and write $g, U^{\mathcal{C}}, b^{\mathcal{C}}$ instead of $g_l, U_l^{\mathcal{C}}, b_l^{\mathcal{C}}$.

Stage 1: Geometric Straightening. The goal of Stage 1 is to learn the straightening map $g(\cdot)$, where we denote the parameters of $g(\cdot)$ as θ_g. We introduce an auxiliary concept-specific orthonormal basis $\tilde{U}^{\mathcal{C}} \in \mathbb{R}^{d \times r}$ for each concept $\mathcal{C}$, with $(\tilde{U}^{\mathcal{C}})^\top \tilde{U}^{\mathcal{C}} = I_r$. In Stage 1, these $\tilde{U}^{\mathcal{C}}$ variables are only used to define the training signal for g. In Stage 2, we freeze g and learn the final clamp parameters $(U^{\mathcal{C}}, b^{\mathcal{C}})$, initializing $U^{\mathcal{C}}$ with the $\tilde{U}^{\mathcal{C}}$ learned in Stage 1.

Let $\mathscr{C}_{\text{train}}$ be the set of concepts used to train the straightening map g. For each concept $\mathcal{C} \in \mathscr{C}_{\text{train}}$, we use prompt pairs $\mathcal{P}_{\mathcal{C}} = \{(x_i^{\mathcal{C}}, x_i)\}_{i=1}^{N_{\mathcal{C}}}$ from the induction and suppression datasets. For each prompt pair, we record hidden states at the intervention site, as $h_i^{\mathcal{C}}$ and h_i. We apply the straightening map $g(\cdot)$ and define a normalized concept direction in the straightened space: $\frac{g(h_i^{\mathcal{C}}) - g(h_i)}{\|g(h_i^{\mathcal{C}}) - g(h_i)\|_2}$. We train g using a composite objective designed to facilitate linear control. Specifically, we require the concept directions $\{\frac{g(h_i^{\mathcal{C}}) - g(h_i)}{\|g(h_i^{\mathcal{C}}) - g(h_i)\|_2}\}_i$ to concentrate within a linear subspace. To achieve this, we define $\Pi_{\mathcal{C}}^{\perp} := I - \tilde{U}^{\mathcal{C}}(\tilde{U}^{\mathcal{C}})^\top$ as the orthogonal projector onto the complement of $\text{span}(\tilde{U}^{\mathcal{C}})$ and minimize the residual energy outside the subspace:

$$\mathcal{L}_{\text{align}} = \sum_{\mathcal{C} \in \mathscr{C}_{\text{train}}} \sum_{i=1}^{i=N_{\mathcal{C}}} \left\| \Pi_{\mathcal{C}}^{\perp} \left(\frac{g(h_i^{\mathcal{C}}) - g(h_i)}{\|g(h_i^{\mathcal{C}}) - g(h_i)\|_2} \right) \right\|_2^2. \tag{3}$$

To ensure the straightening map preserves the model's general capabilities, we incorporate an SFT regularization term. We define $h^{\neg \mathcal{C}} = g^{-1}\Big((I - \tilde{U}^{\mathcal{C}}(\tilde{U}^{\mathcal{C}})^\top)\, g(h) \Big)$ as the concept-ablated hidden state at the current intervention site. We then use this intervened state to compute SFT loss on a generic dataset $\mathcal{D}_{\text{sft}}$:

$$\mathcal{L}_{\text{reg}} = \sum_{\mathcal{C} \in \mathscr{C}_{\text{train}}} \sum_{j=1}^{j=N_{\text{sft}}} \left[-\frac{1}{|y_j|} \log p_0(y_j \mid x_j, h \leftarrow h^{\neg \mathcal{C}}) \right]. \tag{4}$$

This regularizer encourages the straightening map to preserve information needed for general tasks when the concept subspace is ablated. The final Stage 1 objective combines these terms:

$$\begin{aligned} \min_{\theta_g, \{\tilde{U}^{\mathcal{C}}\}_{\mathcal{C} \in \mathscr{C}_{\text{train}}}} \quad & \mathcal{L}_{\text{align}} + \lambda \mathcal{L}_{\text{reg}} \\ \text{s.t.} \quad & (\tilde{U}^{\mathcal{C}})^\top \tilde{U}^{\mathcal{C}} = I_r. \end{aligned} \tag{5}$$

We optimize $\tilde{U}^{\mathcal{C}}$ using the orthogonalization parameterization described in Section 3.3. For each pair of prompts, we record hidden states in each intervened layer and token position. We compute this objective jointly and average at these sites when training the full set $\{g_l\}$. After Stage 1, we freeze the straightening map and move to Stage 2.

[1] This is geometrically equivalent to directional ablation (Arditi et al., 2024) in the straightened coordinates.

Table 1: We compare SAC against training-free and training-based baselines on Qwen-2.5-7B. We report the Attack Success Rate (ASR) for in-distribution (ID) illegal instructions and out-of-distribution (OOD) jailbreak prompts. Lower ASR indicates better defense.

Paradigm	Method	ID ASR	OOD ASR (↓)						
		Illegal	JChat	Cipher	Code	Comp	PAIR	In-the-Wild	Avg
Training-free	Vanilla	7.5	50.5	20.0	66.0	12.0	68.8	27.0	40.7
	ACE	4.0	29.5	4.0	8.0	2.0	6.3	14.5	10.7
	CAST	2.0	31.5	25.5	78.0	8.0	81.3	10.0	39.1
Training-based	SV-1	0.0	8.0	0.0	47.5	1.0	46.9	10.5	19.0
	LinEAS	0.0	9.0	0.0	49.5	0.0	34.4	10.0	17.2
	ReFT	2.0	15.5	6.0	50.5	4.0	54.7	10.3	23.5
	LoFiT	0.0	8.5	2.0	45.0	0.5	35.9	7.0	16.5
	SAC	0.0	5.0	0.0	45.5	0.0	31.3	5.3	14.5

Stage 2: Concept-specific Clamping. In Stage 2, with g fixed, we learn the concept-specific clamp parameters $(U^{\mathcal{C}}, b^{\mathcal{C}})$ for concept $\mathcal{C} \in \mathscr{C}_{\text{train}}$. Following (Wu et al., 2025b), we use a pairwise logistic preference objective to distinguish between concept induction and concept suppression behaviors.

Concept Induction. When $b \neq 0$, we want the model to prefer the concept triggered response $y_i^{\mathcal{C}}$ over the neutral response y_i. We define the pairwise logits $\Delta^{+}_{\text{ind}}(i) = s_{\text{ind}}(x_i, y_i^{\mathcal{C}}) - s_{\text{ind}}(x_i, y_i)$ and $\Delta^{-}_{\text{ind}}(i) = s_{\text{supp}}(x_i, y_i) - s_{\text{supp}}(x_i, y_i^{\mathcal{C}})$. The preference loss is defined as $\mathcal{L}_{\text{ind}} = -\frac{1}{N_{\mathcal{C}}} \sum_{i=1}^{N_{\mathcal{C}}} \left[\log \sigma(\Delta^{+}_{\text{ind}}(i)) + \log \sigma(\Delta^{-}_{\text{ind}}(i))\right]$. The second term encourages the suppression setting to reverse the preference, sharpening the separation between induction and suppression behaviors. We optimize concept induction on the neutral prompts x_i so that SAC turns on concept $\mathcal{C}$ even when the prompt contains no explicit trigger.

Concept Suppression. When $b = 0$, we prioritize the suppression of the concept. Given a concept trigger prompt $x_i^{\mathcal{C}}$, we want the concept suppressed response $y_i^{\neg\mathcal{C}}$ to be preferred over the concept triggered response $y_i^{\mathcal{C}}$. Define $\Delta^{+}_{\text{supp}}(i) = s_{\text{supp}}(x_i^{\mathcal{C}}, y_i^{\neg\mathcal{C}}) - s_{\text{supp}}(x_i^{\mathcal{C}}, y_i^{\mathcal{C}})$ and preference loss $\mathcal{L}_{\text{supp}} = -\frac{1}{N_{\mathcal{C}}} \sum_{i=1}^{N_{\mathcal{C}}} \log \sigma(\Delta^{+}_{\text{supp}}(i))$. We also include an SFT-style term to directly increase the likelihood of the desired responses under the concept induction or suppression setting. The overall objective is the combination $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{control}} + \mathcal{L}_{\text{SFT}}$, where $\mathcal{L}_{\text{control}}$ is $\mathcal{L}_{\text{ind}}$ for induction or $\mathcal{L}_{\text{supp}}$ for suppression.

5 Experiments

5.1 Experiments Setup

Models and Datasets. We used Qwen-2.5-7B and Gemma-2-9B as the base models for our experiments. For concept induction, we employ the Concept500 dataset from AxBench (Wu et al., 2025a), which consists of instructions paired with both concept-induced and neutral responses. We partition the instructions for each concept into training, validation, and testing sets using a 8:1:1 ratio. For concept suppression, we conducted experiments to suppress the harmful concept. We construct a training dataset that combines 60,000 benign instructions from Evol-Instruct (Xu et al., 2023) with 10,000 illegal instructions from Beavertails (Ji et al., 2023), maintaining a 6:1 ratio.

For the evaluation of concept induction, we use the AxBench Concept500 test split. We generate up to 512 tokens for each test instruction. We selected hyperparameters on the validation split only and evaluated only on the test split. Harmful Concept Suppression includes both in-distribution (ID) illegal instructions and out-of-distribution (OOD) jailbreak prompts. For ID illegal instructions, we test illegal instructions derived from Do-Not-Answer (Wang et al., 2024), HarmBench (Mazeika et al., 2024), and WildChat (Zhao et al., 2024). For OOD jailbreak prompts, we evaluate the safety of the model against six distinct attack types: JailbreakingChat (Walkerspider, 2022), SelfCipher (Yuan et al., 2024), Code Attack (Ren et al., 2024), Completion Attack (Liu et al., 2024), PAIR (Chao et al., 2023), and the jailbreak version of WildChat toxic prompts (Zhao et al., 2024). Our intervention modules are applied to the residual stream. For AxBench evaluation, we attach interventions to layer 20, following the AxBench recipe. For the concept suppression task, following Wu et al. (2024), we attach interventions at every transformer layer and every prompt-token position during prefill to

(a) Probe AUC

Method	AUC ↑
Vanilla	0.831
SAC (HF)	0.963
SAC (NICE)	**0.970**

(b) Shared vs. concept-specific straightening map

Variant	Qwen-2.5-7B	Gemma-2-9B
Shared	1.072	1.034
Concept-specific	**1.157**	**1.052**

(c) Intervention stability under increasing intervention layers

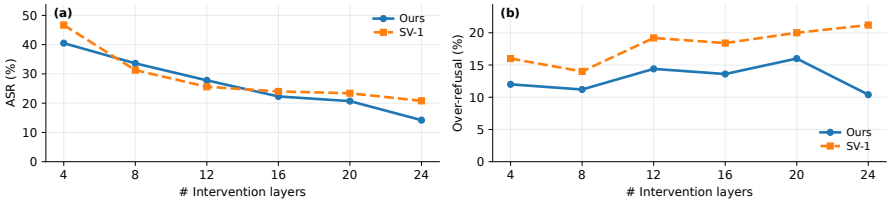


Figure 2: **Diagnostics and ablations for SAC.** (a) Linear probe AUC for classifying jailbreak vs. benign prompts. (b) Ablating the scope of the straightening map on AxBench. (c) Attack Success Rate and over-refusal rate as the number of intervened layers varies from 4 to 24.

Table 2: We report the harmonic mean of the Concept Score, Instruct Score, and Fluency Score on the AxBench Concept500 dataset. We compare SAC against state-of-the-art training-based and training-free baselines. Higher is better. † means results are taken from Wu et al. (2025a).

		Harmonic score (↑)	
Method	**Obj.**	**Qwen-2.5-7B**	**Gemma-2-9B**
LoFiT	RePS	0.914	0.851
SV-1	RePS	0.955	0.910
ReFT	RePS	0.857	0.823
LoRA	RePS	0.874	0.789
DiffMean	NA	0.451	0.322†
HyperSteer	Lang.	NA	1.091†
SAE	Lang.	NA	0.546†
SAC (NICE)	Two-stage	1.072	1.034
+ *Code Sweep*	Two-stage	**1.201**	**1.120**
SAC (HF)	Two-stage	1.034	1.025
+ *Code Sweep*	Two-stage	1.157	1.079

prevent the downstream re-emergence of harmful concepts. Because we intervene only during prefill, the method remains compatible with KV caching in modern LLM inference engines.

Metrics. For concept induction, following Wu et al. (2025a), we report the harmonic mean of three metrics: the Concept Score, the Instruct Score, and the Fluency Score. A judge LLM rates each score on a scale of 0 to 2 following structured prompts (see Appendix G.1). For harmful concept suppression, we report the Attack Success Rate (ASR), where a lower score indicates a stronger defense. We distinguish between ID ASR and OOD ASR to assess the faithfulness of the intervention. For PAIR and CodeAttack, we adopt standard guidelines (Wang et al., 2025) for inference. We compare our proposed method against both training-based and training-free representation intervention techniques. The training-based baselines include Representation Fine-Tuning (ReFT) (Wu et al., 2024), Localized Fine-tuning (LoFiT) (Yin et al., 2024), RePS (Wu et al., 2025b) and LinEAS (Rodriguez et al., 2025). We also evaluate training-free intervention methods: Difference in Means (DiffMean), HyperSteer (Sun et al., 2025), SAE-based Steering (SAE) (Lieberum et al., 2024), Affine Concept Editing (ACE) (Marshall et al., 2024), and Conditional Activation Steering (CAST) (Lee et al., 2024).

5.2 Main Result

As shown in Tables 2 and Table 1, SAC demonstrates superior performance in both concept induction and harmful concept suppression tasks. For concept induction on the AxBench dataset (Table 2),

SAC achieves the highest harmonic scores. Notably, when combined with the code sweep (described in Section 5.3), it significantly outperforms state-of-the-art representation intervention baselines such as LoFiT and ReFT, demonstrating its ability to faithfully elicit target concepts while maximally preserving general instruction-following capabilities and text fluency for an optimal control–utility trade-off. Furthermore, the harmful concept suppression results in Table 1 highlight SAC's exceptional robustness against distribution shifts. Whether evaluated on in-distribution (ID) illegal instructions or a diverse set of out-of-distribution (OOD) jailbreak prompts (e.g., JChat, Cipher, Code), SAC yields an average OOD ASR of 14.5% on Qwen-2.5-7B and comprehensively outperforming training-based defense methods. Among training-free methods, ACE achieves the lowest ASR. However, it indiscriminately applies the refusal vector to all prompts. In contrast, SAC dynamically performs concept-specific steering based on the harmful concepts identified in each prompt, achieving comparable ASR while lower over-refusal (see Sec. 5.3). We have additional experiments on SAC optimal configuration in Appendix D.

5.3 Ablations

Code Strength Sweep. In our method, concept control is achieved by clamping the straightened concept coordinates to a target code b. To expose this as an explicit control, we reparameterize $b = \alpha\,\omega$, where $\|\omega\|_2 = 1$. The ω fixes the concept direction and α controls the intervention strength. We keep ω fixed and vary α at inference. We select α on the validation split of AxBench for each concept. For each baseline with an inference-time steering strength, we also use its official AxBench search range and tune on the same validation examples to have a fair comparison. Thanks to the idempotent clamp, this sweep can further achieve significant improvements as in Table 2.

Straightening Architecture. We ablate the parameterization of the straightening map by comparing (i) NICE and (ii) non-linear Householder flow (HF). Both choices preserve the key requirement of exact invertible coupling flows. NICE provides high expressivity with translation networks and permutations. HF is designed as a more efficient geometric preconditioner, with minimal overhead compared to baseline intervention methods, such as ReFT. HF achieves comparable downstream harmonic means in Table 2, suggesting moderate non-linear straightening is sufficient. We also ablate the scope of the straightening map g in Stage 1 by comparing a concept-specific variant, which trains an independent g_c for each target concept c, against our default shared variant. As shown in Fig. 2, training a separate g_c yields no significant improvement over the shared-g approach. Given its comparable performance and substantially better scalability, we adopt the shared-g variant.

Intervention Stability. To empirically test the intervention stability under increasing intervention density, we sweep the number of intervened layers. For the layer number sweep, we vary the number of intervened layers in $\{4, 8, 12, 16, 20, 24\}$. We evaluate the attack success rate of jailbreak prompts and over-refusal rate on benign prompts from XsTest. As in Fig. 2, our method can decrease the attack success rate with denser intervention without a significant increase in over-refusal.

Probe Diagnostics. To verify whether the straightening map achieves its intended geometric effect, we collect the residual-stream hidden states for the last prompt token at layer 15 of Qwen-2.5-7B. We use 2000 jailbreak and benign prompts from WildGuard (Han et al., 2024), and then fit a linear probe. For the base model, we train and evaluate the probe in the native representation space. For our method, we additionally apply the learned straightening map and probe the straightened coordinates. As in Fig. 2, AUC on the test split prompts from WildGuard improves substantially after straightening. This provides quantitative support for performing subspace clamping in straightened space.

6 Conclusions

In this paper, we propose SAC, a representation intervention framework for faithful and robust concept control in LLMs. To address the "hydra effect" and the non-linear geometry of latent concepts that hinder existing methods, SAC employs an invertible straightening map to concentrate concept variations into a linear subspace, followed by an idempotent projection clamp. This geometric design enables dense, multi-layer interventions. Supported by a parameter-efficient, two-stage training strategy, our evaluations on concept induction and harmful concept suppression demonstrate that SAC delivers a superior control–utility trade-off and enhanced robustness.

LLM USAGE

In this work, we used large language models (LLMs) for two purposes. First, LLMs were used to assist with manuscript preparation by editing author-written text for grammar, fluency, clarity, and consistency of academic tone. This assistance was limited to language polishing and did not determine the scientific claims or conclusions of this work. Second, LLMs were used to generate synthetic responses for dataset construction. The corresponding data format, construction procedure, and prompt templates are described in Sec. 4 and Appendix G. All LLM-assisted manuscript text and generated data were reviewed by the authors. The authors take full responsibility for the final content of this work, including its text, claims, results, and artifacts.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024. (Cited on pages 1 and 6)

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36:66044–66063, 2023. (Cited on page 3)

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, et al. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*, 2026. (Cited on page 1)

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023. (Cited on page 7)

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029, 2024. (Cited on page 1)

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. (Cited on page 2)

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. (Cited on page 5)

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019. (Cited on page 16)

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas F. Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*, 2023. (Cited on pages 1 and 2)

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR, 2024. (Cited on pages 1 and 2)

Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, 2019. (Cited on page 3)

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. (Cited on page 1)

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. (Cited on page 1)

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in neural information processing systems*, 37:8093–8131, 2024. (Cited on pages 1 and 9)

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668, 2023. (Cited on page 2)

Nicholas J Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986. (Cited on page 5)

Lei Huang, Li Liu, Fan Zhu, Diwen Wan, Zehuan Yuan, Bo Li, and Ling Shao. Controllable orthogonalization in training dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6429–6438, 2020. (Cited on page 5)

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023. (Cited on page 7)

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*, 2024. (Cited on page 8)

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024. (Cited on pages 2 and 8)

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Jwpw4qKkb. (Cited on page 7)

Jinqi Luo, Tianjiao Ding, Kwan H Chan, Darshan Thaker, Aditya Chattopadhyay, Chris Callison-Burch, and René Vidal. Pace: Parsimonious concept engineering for large language models. *Advances in Neural Information Processing Systems*, 37:99347–99381, 2024. (Cited on page 1)

Thomas Marshall, Adam Scherlis, and Nora Belrose. Refusal in llms is an affine function. *arXiv preprint arXiv:2411.09003*, 2024. (Cited on page 8)

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, 2024. (Cited on page 7)

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023. (Cited on pages 2 and 3)

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025. (Cited on pages 2 and 3)

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, et al. Mib: A mechanistic interpretability benchmark. *arXiv preprint arXiv:2504.13151*, 2025. (Cited on pages 1 and 2)

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023. (Cited on pages 2 and 4)

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024. (Cited on page 2)

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7237–7256, 2020. (Cited on page 3)

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pages 18400–18421. PMLR, 2022a. (Cited on page 3)

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6034–6055, 2022b. (Cited on pages 2 and 3)

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. Exploring safety generalization challenges of large language models via code. *arXiv preprint arXiv:2403.07865*, 2024. (Cited on page 7)

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024. (Cited on pages 1 and 3)

Pau Rodriguez, Michal Klein, Eleonora Gualdoni, Valentino Maiorca, Arno Blaas, Luca Zappella, Xavier Suau, et al. Lineas: End-to-end learning of activation steering with a distributional loss. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. (Cited on pages 1, 3 and 8)

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025. (Cited on page 2)

Michael Spivak. *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*. CRC press, 2018. (Cited on pages 4 and 15)

Jiuding Sun, Sidharth Baskaran, Zhengxuan Wu, Michael Sklar, Christopher Potts, and Atticus Geiger. Hypersteer: Activation steering at scale with hypernetworks. *arXiv preprint arXiv:2506.03292*, 2025. (Cited on page 8)

Hieu M Vu and Tan M Nguyen. Angular steering: Behavior control via rotation in activation space. *arXiv preprint arXiv:2510.26243*, 2025. (Cited on page 1)

Walkerspider. DAN is my new friend., https://old.reddit.com/r/ChatGPT/comments/zlcyr9/dan_is_my_new_friend/, 2022. (Cited on page 7)

Haoyu Wang, Zeyu Qin, Li Shen, Xueqian Wang, Minhao Cheng, and Dacheng Tao. Leveraging reasoning with guidelines to elicit and utilize knowledge for enhancing safety alignment. *arXiv preprint arXiv:2502.04040*, 2025. (Cited on page 8)

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022. (Cited on page 3)

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: Evaluating safeguards in LLMs. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 896–911, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.61. (Cited on page 7)

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in neural information processing systems*, 36:78205–78226, 2023. (Cited on pages 1 and 2)

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024. (Cited on pages 1, 3, 4, 7 and 8)

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Daniel Jurafsky, Christopher D. Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025a. (Cited on pages 7 and 8)

Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D Manning, and Christopher Potts. Improved representation steering for language models. *arXiv preprint arXiv:2505.20809*, 2025b. (Cited on pages 3, 7 and 8)

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, et al. Sorry-bench: Systematically evaluating large language model safety refusal. International Conference on Learning Representations (ICLR), 2025. (Cited on page 1)

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023. (Cited on page 7)

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. (Cited on page 1)

Hongzheng Yang, Yongqiang Chen, Zeyu Qin, Tongliang Liu, Chaowei Xiao, Kun Zhang, and Bo Han. Does representation intervention really identify desired concepts and elicit alignment? *arXiv preprint arXiv:2505.18672*, 2025b. (Cited on page 3)

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations. *Advances in Neural Information Processing Systems*, 37:9474–9506, 2024. (Cited on pages 1, 3 and 8)

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MbfAK4s61A. (Cited on page 7)

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM. (Cited on page 7)

A Broader Impacts

SAC is intended to improve faithful and robust concept-level control in language models. Its positive societal impact is most direct in safety-oriented settings: by suppressing harmful concepts and improving robustness to distribution shifts such as jailbreak prompts, SAC may help reduce harmful compliance while preserving general model utility. The same framework may also support benign customization, such as inducing desired writing styles, personas, or domain framing without full model fine-tuning.

At the same time, SAC is a dual-use representation intervention technique. Although our primary motivation is safety and faithful control, concept induction could be misused to amplify harmful, deceptive, biased, or manipulative behaviors if the target concept and training data are chosen maliciously. For high-risk concepts, access to trained intervention modules may need to be controlled, and deployments should include monitoring, red-teaming, and fallback safeguards.

B Limitations and Future Work

One limitation of SAC is that its current dense deployment does not adaptively select intervention layers. In the harmful-concept suppression setting, we intervene across all transformer layers during prefill in order to reduce the re-emergence of the target concept through alternative computational pathways. While this dense strategy is useful for robustness, it also introduces additional prefill-time computation and may apply unnecessary interventions at layers that are weakly related to the target concept. A promising direction for future work is adaptive layer selection, where the intervention sites are chosen according to concept-specific sensitivity, validation-time control–utility trade-offs, or runtime constraints. Such an adaptive strategy could preserve most of the robustness benefits of dense deployment while reducing computational overhead and mitigating potential utility degradation caused by intervening at irrelevant layers.

C Additional Experimental Details

Table 4 highlights a key trade-off between (i) where computation is paid (prefill vs. decode) and (ii) how costs scale with the number of intervened concepts. LoFiT is parameter-light, but its intervention is applied during decoding, so latency grows with T_g and becomes increasingly unfavorable for long-form generation. At the other extreme, merged LoRA adds no runtime FLOPs, but scales poorly in per-concept storage, since each new concept requires storing a full set of low-rank weights across layers/projections. LoReFT avoids decode-time overhead under prompt-only application, yet still requires a comparatively larger per-concept memory footprint.

Our method shifts most expressivity into a shared straightening map g that is amortized across concepts, while keeping only a small per-concept clamp. This yields two practical advantages: (1) substantially better scaling to many concepts via reduced per-concept parameters, and (2) decode-free overhead under KV-cache-compatible prompt-only use, making inference cost largely insensitive to T_g. Finally, SAC compute is tunable through the choice of g, allowing a controlled trade between nonlinearity and overhead while preserving these scaling benefits. C_g captures the cost of evaluating g and g^{-1}. For a K_f-step Householder flow, $C_g \approx 8K_f d$, giving compute comparable to LoReFT, while still preconditioning the representation geometry.

D Additional Experiments

Matched comparison protocol. We isolate the effects of straightening, and clamping. All variants use the same training examples, and total Stage-1 and Stage-2 optimization updates. Native-space Clamp sets $g = I$ and retains the projection operator. Straightened Additive uses the learned g but replaces the projection with $q' = q + \alpha u$. Native-space Clamp obtains 0.941 and 0.897 harmonic score on Qwen-2.5-7B and Gemma-2-9B respectively. Straightened Additive changes these values to 0.987 and 0.960, while full SAC obtains 1.072 and 1.034. Relative to Native-space Clamp, straightening improves harmonic score by 0.131 on Qwen-2.5-7B. Relative to Straightened Additive, projection clamping improves the harmonic score by 0.085.

Table 3: Component ablations for SAC. We isolate the coordinate map, intervention operator, and sharing strategy on AxBench. Higher is better for harmonic score.

Variant	Straightening	Operator	Map scope	Qwen HM↑	Gemma HM↑
Native-space Clamp	No	Projection	–	0.941	0.897
Straightened Additive	Yes	Additive	Shared	0.987	0.960
SAC (shared)	Yes	Projection	Shared	1.072	1.034
SAC (concept-specific)	Yes	Projection	Per concept	1.157	1.052

Table 4: We calculate the compute and memory overhead for our method and baselines. We count a fused multiply-add as 2 FLOPs (a scalar add as 1 FLOP). The symbol definitions can be found in Table 5

	FLOPs overhead		**Memory overhead (params)**	
Method	**Prefill (T_p)**	**Decode (T_g)**	**Shared**	**Per concept/task**
LoRA	0	0	0	$2\,L\,N_{\text{in}}\,d\,r$
LoReFT	$6\,T_p\,L_{\text{int}}\,d\,r$	0	0	$L_{\text{int}}(2dr + r)$
LoFiT	$T_p\,K_h\,d_h$	$T_g\,K_h\,d_h$	0	$K_h\,d_h$
Ours	$T_p\,L_{\text{int}}\,(C_g + 4dr)$	0	$\text{Params}(g)$	$L_{\text{int}}(dr + r)$

E Straightening Map Architectures

E.1 Straightening Map Existence

Proposition E.1 (Existence of a Local Straightening Map; (Spivak, 2018, Theorem 2-13))**.** *Let $\varphi : \mathbb{R}^d \to \mathbb{R}^r$ be a continuously differentiable function that estimates the concept from the hidden state. Assume that the Jacobian of φ has constant rank r in a neighborhood of a point $h_0 \in \mathbb{R}^d$. Then there exists a local diffeomorphism $g : \mathbb{R}^d \to \mathbb{R}^d$ defined on a neighborhood of h_0 such that in the straightened coordinates $q = g(h) \in \mathbb{R}^d$, the concept function has the form $\varphi(h) = L(q_1, \ldots, q_r)$ where $L : \mathbb{R}^r \to \mathbb{R}^r$ is an invertible linear map and $(q_1, \ldots, q_r)$ are the first r coordinates of $q \in \mathbb{R}^d$.*

This is a local existence statement near a regular point. In the main paper, we use it as a motivation for learning an amortized preconditioner g on the data manifold, not as a claim that one shared global chart exactly straightens all contexts or all concepts.

E.2 NICE

A NICE block uses additive couplings. Let $m \in \{0, 1\}^d$ be a binary mask and P a fixed permutation matrix. With two small translation networks T_1 and T_2, the forward map is

$$\begin{aligned} h'_{\bar{m}} &= h_{\bar{m}} + T_1(h_m), \\ h'_m &= h_m, \\ h''_m &= h'_m + T_2(h'_{\bar{m}}), \\ h''_{\bar{m}} &= h'_{\bar{m}}, \\ h^{\text{out}} &= Ph'', \end{aligned}$$

and the inverse subtracts the learned translations in reverse order after undoing the permutation. Stacking a small number of blocks with alternating masks and fixed permutations yields an expressive and exactly invertible g. NICE has unit Jacobian determinant per block and often trains robustly. An ActNorm layer $w \mapsto \gamma \odot w + \beta$ with data-dependent initialization on the first batch is placed before the first coupling block to stabilize training.

E.3 Non-Linear Householder Flows

We implement the straightening map g using a composition of Householder reflections. A standard Householder reflection is an orthogonal transformation that reflects a vector about a hyperplane

Table 5: Symbol definitions used in the compute/memory overhead analysis.

Symbol	Meaning
L	total number of transformer layers in the backbone
$\mathcal{L}_{\text{int}}$	set of intervened layers
L_{int}	number of intervened layers, $L_{\text{int}} = \lvert\mathcal{L}_{\text{int}}\rvert$
T_p	number of prompt tokens processed in prefill
T_g	number of generated tokens processed in decode
$\mathcal{P}$	token positions intervened per layer during prefill
$\lvert\mathcal{P}\rvert$	number of intervened positions (typically $\lvert\mathcal{P}\rvert = T_p$)
d	residual-stream width (hidden size)
H	number of attention heads
d_h	per-head dimension, $d_h = d/H$
r	clamp subspace dimension (rank)
K_h	number of heads edited by LoFiT
K_f	number of steps in the Householder-flow straightener
C_g	FLOPs to compute $g(\cdot)$ and $g^{-1}(\cdot)$ once for one d-dimensional vector
$\text{Params}(g)$	parameter count of the shared straighteners $\{g_l\}_{l\in\mathcal{L}_{\text{int}}}$

passing through the origin. Let $u_k \in \mathbb{R}^d$ be a learnable parameter vector. We define the unit normal vector $v_k = u_k/\|u_k\|_2$. The Householder matrix $H_k \in \mathbb{R}^{d\times d}$ associated with v_k is defined as:

$$H_k = I - 2v_k v_k^T,$$

Since H_k is orthogonal, the transformation is isometric (preserving Euclidean distances) and volume-preserving ($|\det(H_k)| = 1$). We construct the map g as a sequence of K such reflections, where $K \ll d$. Let $h^{(0)} = h$ be the input vector. The forward pass is defined recursively for $k = 1, \ldots, K$:

$$h^{(k)} = H_k h^{(k-1)} = h^{(k-1)} - 2v_k(v_k^T h^{(k-1)}),$$

We avoid explicitly constructing the matrix H_k. By computing the inner product $v_k^T h^{(k-1)}$ first, the operation reduces to vector-vector arithmetic with a computational complexity of $\mathcal{O}(d)$ per step, resulting in a total complexity of $\mathcal{O}(Kd)$. The inverse map g^{-1} is efficient because Householder matrices are symmetric and orthogonal, meaning they are their own inverses ($H_k^{-1} = H_k^T = H_k$). To invert the sequence, we apply the reflections in reverse order from $k = K$ down to 1.

While standard Householder reflections are computationally efficient, they are strictly linear operations limited to rigid rotations and reflections. This restricts the model to learning a global rotation of the latent manifold. To introduce non-linearity while retaining the $\mathcal{O}(d)$ efficiency and geometric interpretability, we generalize the reflection coefficient.

We observe that the term -2 in the linear update acts as a fixed scalar multiplier on the projection of the data onto the normal vector v_k. To allow for complex deformations of the manifold, we replace this fixed scalar with a learnable, non-linear function. Let $s_{k-1} = v_k^T h^{(k-1)}$ be the scalar projection of the input onto the unit normal vector v_k. We introduce a learnable, strictly monotonic and thus invertible scalar function $\psi_k : \mathbb{R} \to \mathbb{R}$. The Non-Linear Householder transformation is defined as:

$$h^{(k)} = h^{(k-1)} + (\psi_k(s_{k-1}) - s_{k-1})v_k.$$

Geometrically, this operation decomposes the input vector into two orthogonal components: one parallel to v_k and one orthogonal to v_k. The orthogonal component remains unchanged, while the parallel component is mapped from s_{k-1} to $s_k = \psi_k(s_{k-1})$. If we choose $\psi_k(s) = -s$, we recover the exact linear Householder reflection, as $(-s - s) = -2s$. This formulation constitutes a normalizing flow. We parameterize ψ_k using rational quadratic splines. We enforce monotonicity using the standard constrained parameterization for rational quadratic splines (Durkan et al., 2019).

E.4 Necessity of Non-linear Straightening

Proposition E.2 (Insufficiency of ReFT)**.** *Assume that the concept code $c = \varphi(h)$ is a non-linear function of h. Let $g(h) = Ah$ be any invertible linear map. Then, in the straightened coordinates $q = Ah$, the concept remains a non-linear function of q. That is, there exists no fixed matrix U such that $\varphi(h) = U^\top q$ for all h.*

Proof of Proposition E.2. Because g is linear and invertible, we can write $h = A^{-1}q$. Substituting into φ gives

$$c = \varphi(h) = \varphi(A^{-1}q) \equiv \tilde{\varphi}(q),$$

where $\tilde{\varphi} = \varphi \circ A^{-1}$.

We need to show that $\tilde{\varphi}$ is non-linear. Since φ is non-linear by assumption, there exist vectors $h_1, h_2 \in \mathbb{R}^d$ and scalars $\alpha, \beta \in \mathbb{R}$ such that

$$\varphi(\alpha h_1 + \beta h_2) \neq \alpha\varphi(h_1) + \beta\varphi(h_2), \tag{1}$$

Define $q_1 = Ah_1$ and $q_2 = Ah_2$. Because A is linear,

$$\alpha q_1 + \beta q_2 = A(\alpha h_1 + \beta h_2),$$

Now evaluate $\tilde{\varphi}$ at the linear combination:

$$\begin{aligned}\tilde{\varphi}(\alpha q_1 + \beta q_2) &= \varphi\big(A^{-1}(\alpha q_1 + \beta q_2)\big) \\ &= \varphi\big(\alpha A^{-1} q_1 + \beta A^{-1} q_2\big) \\ &= \varphi(\alpha h_1 + \beta h_2),\end{aligned} \tag{2}$$

On the other hand, if $\tilde{\varphi}$ were linear, we would have

$$\alpha\tilde{\varphi}(q_1) + \beta\tilde{\varphi}(q_2) = \alpha\varphi(A^{-1}q_1) + \beta\varphi(A^{-1}q_2) = \alpha\varphi(h_1) + \beta\varphi(h_2), \tag{3}$$

But Eq. equation 2 and Eq. equation 3 together would imply

$$\varphi(\alpha h_1 + \beta h_2) = \alpha\varphi(h_1) + \beta\varphi(h_2).$$

which contradicts Eq. equation 1. Hence $\tilde{\varphi}$ cannot be linear. Consequently, there is no matrix U such that $\tilde{\varphi}(q) = U^\top q$ for all q, because the right-hand side is a linear function. □

F Constraint satisfaction, fixed points, idempotence, and drift removal

Proposition F.1 (Constraint satisfaction, fixed points, idempotence, and drift removal)**.** *Assume $g : \mathbb{R}^d \to \mathbb{R}^d$ is invertible and $U \in \mathbb{R}^{d\times r}$ has orthonormal columns, i.e., $U^\top U = I_r$. For a desired concept code $b \in \mathbb{R}^r$, define the affine clamp in the straightened coordinates by $\Phi_{(U,b)}(q) := (I - UU^\top)q + Ub,\ q \in \mathbb{R}^d$, and define the corresponding intervention $f(h) := g^{-1}\big(\Phi_{(U,b)}(g(h))\big)$. Let $\mathcal{H}_{g,U,b} := \{h \in \mathbb{R}^d : U^\top g(h) = b\}$ be the target concept-constraint set in the native coordinates. Then, for any $h \in \mathbb{R}^d$, letting $q := g(h)$ and $\delta := U^\top q - b$, the following properties hold:*

- *(i)* ***Constraint satisfaction.*** *The intervened straightened representation satisfies $U^\top\Phi_{(U,b)}(q) = U^\top g(f(h)) = b$, and hence it implies that $f(h) \in \mathcal{H}_{g,U,b}$.*
- *(ii)* ***Fixed points and idempotence.*** *The fixed-point set of f is exactly $\mathcal{H}_{g,U,b}$, that is, $f(h) = h$ if and only if $U^\top g(h) = b$. Moreover, $\Phi_{(U,b)}(\Phi_{(U,b)}(q)) = \Phi_{(U,b)}(q), f(f(h)) = f(h)$.*
- *(iii)* ***Drift removal.*** *The straightened representation after intervention is $g(f(h)) = g(h) - U\delta$. Equivalently, the clamp removes the concept-coordinate residual $\delta = U^\top g(h) - b$ while preserving the orthogonal component: $(I - UU^\top)g(f(h)) = (I - UU^\top)g(h)$.*

Proof. Let $q := g(h)$ and $\delta := U^\top q - b$. Observe that

$$\begin{aligned}\Phi_{(U,b)}(q) &= (I_d - UU^\top)q + Ub \\ &= q - UU^\top q + Ub \\ &= q - U\big(U^\top q - b\big) \\ &= q - U\delta.\end{aligned} \tag{4}$$

By the definition of f and the invertibility of g, we have

$$g(f(h)) = g\big(g^{-1}\big(\Phi_{(U,b)}(g(h))\big)\big) = \Phi_{(U,b)}(g(h)) = \Phi_{(U,b)}(q).$$

Part (i). By equation equation 4, we can compute

$$\begin{aligned} U^\top \Phi_{(U,b)}(q) &= U^\top \big((I_d - UU^\top)q + Ub\big) \\ &= U^\top q - U^\top UU^\top q + U^\top Ub \\ &= U^\top q - I_r U^\top q + I_r b = b. \end{aligned}$$

Since $g(f(h)) = \Phi_{(U,b)}(q)$, it follows that $U^\top g(f(h)) = U^\top \Phi_{(U,b)}(q) = b$. Therefore, by the definition of $\mathcal{H}_{g,U,b}$, we have $f(h) \in \mathcal{H}_{g,U,b}$.

Part (ii). We first prove that the fixed-point set of f is exactly $\mathcal{H}_{g,U,b}$. If $h \in \mathcal{H}_{g,U,b}$, then $U^\top g(h) = b$. Since $q = g(h)$, this means $U^\top q = b$, and hence $\delta = 0$. By equation 4, $\Phi_{(U,b)}(q) = q$. Therefore,

$$f(h) = g^{-1}\big(\Phi_{(U,b)}(g(h))\big) = g^{-1}(g(h)) = h,$$

which implies that every $h \in \mathcal{H}_{g,U,b}$ is a fixed point of f. Conversely, suppose $f(h) = h$. Applying g to both sides gives $g(f(h)) = g(h)$. Using $g(f(h)) = \Phi_{(U,b)}(q)$ and $q = g(h)$, we obtain $\Phi_{(U,b)}(q) = q$. Applying $U^\top$ to both sides and using Part (i), we get $U^\top q = U^\top \Phi_{(U,b)}(q) = b$. Hence $U^\top g(h) = b$, so $h \in \mathcal{H}_{g,U,b}$. Therefore, $\mathrm{Fix}(f) = \mathcal{H}_{g,U,b}$.

Next, we prove the property of idempotence. Let $q^+ := \Phi_{(U,b)}(q)$, then by Part (i), $U^\top q^+ = b$. Using the fixed-point property of $\Phi_{(U,b)}$ in the straightened coordinates, $\Phi_{(U,b)}(q^+) = q^+$. Equivalently, $\Phi_{(U,b)}(\Phi_{(U,b)}(q)) = \Phi_{(U,b)}(q)$. For f, we have

$$g(f(f(h))) = \Phi_{(U,b)}(g(f(h))) = \Phi_{(U,b)}(\Phi_{(U,b)}(q)) = \Phi_{(U,b)}(q) = g(f(h)).$$

Since g is invertible, it is injective. Hence $f(f(h)) = f(h)$.

Part (iii). By equation 4 and $g(f(h)) = \Phi_{(U,b)}(q)$, we have

$$g(f(h)) = \Phi_{(U,b)}(q) = q - U\delta = g(h) - U\delta.$$

Thus, the intervention subtracts the concept-coordinate residual $\delta = U^\top g(h) - b$ along the subspace $\mathrm{span}(U)$. Finally, the orthogonal component in the straightened coordinates is preserved:

$$\begin{aligned} (I_d - UU^\top)g(f(h)) &= (I_d - UU^\top)(q - U\delta) \\ &= (I_d - UU^\top)q - (I_d - UU^\top)U\delta \\ &= (I_d - UU^\top)q \\ &= (I_d - UU^\top)g(h), \end{aligned}$$

where we use the fact that

$$(I_d - UU^\top)U = U - U(U^\top U) = U - UI_r = 0.$$

This proves the drift-removal and orthogonal-component preservation claims. □

G Prompt Templates

G.1 Evaluation Prompts

We adopt gpt-oss-20b as the judge model with the following prompts.

Table 6: Hyperparameters for SAC.

Item	Value
Train/validation/test split	8:1:1 for AxBench Concept500
Maximum generation length	512 tokens
Intervention site for AxBench	Residual stream, layer 20
Intervention site for harmful suppression	Residual stream, all transformer layers during prefill
Straightening architectures	NICE and non-linear Householder flow
NICE coupling blocks	2
Concept subspace dimension r	16
Newton–Schulz iterations for orthogonalization	10
Optimizer	AdamW
Stage 1 learning rate	1e−4
Stage 2 learning rate	1e−3
Batch size	32
Training epochs	3
GPU memory	1 * H800 80GB

Concept score

[System]
Please act as an impartial judge and evaluate whether the specified concept is presented in the sentence fragment provided below. Focus solely on whether the concept is clearly incorporated, without regard for grammar or logical coherence.

Begin your evaluation by providing a short explanation of whether the concept is clearly incorporated. Point out references to the concept in the fragment, noting any minimal or awkward incorporations. After providing your explanation, rate the concept's relevance on a scale from 0 to 2, where 0 indicates the concept is not present at all, 1 indicates the concept is somewhat present but minimally or awkwardly incorporated, and 2 indicates the concept is more fully and effectively incorporated, with stronger and more natural integration. Provide your rating using this exact format: "Rating: [[score]]".

[Concept Start]
[Concept goes here]
[Concept End]

[Response Start]
[Sentence goes here]
[Response End]

Instruct score

[System]
Please act as an impartial judge and evaluate whether the sentence fragment provided below is related to the instruction. Focus solely on the degree of relatedness in terms of topic, regardless of grammar, coherence, or informativeness.

Begin your evaluation by providing a brief explanation of whether the sentence is related to the instruction, and point out references related to the instruction. After providing your explanation, rate the instruction relevance on a scale from 0 to 2, where 0 indicates the sentence is unrelated to the instruction, 1 indicates it is somewhat related but only minimally or indirectly relevant in terms of topic, and 2 indicates it is more clearly and directly related to the instruction. Provide your rating using this exact format: "Rating: [[score]]".

[Instruction Start]
[Instruction goes here]
[Instruction End]

[Response Start]
[Sentence goes here]
[Response End]

Fluency score

[System]
Please act as an impartial judge and evaluate the fluency of the sentence fragment provided below. Focus solely on fluency, disregarding its completeness, relevance, coherence with any broader context, or informativeness.

Begin your evaluation by briefly describing the fluency of the sentence, noting any unnatural phrasing, awkward transitions, grammatical errors, or repetitive structures that may hinder readability. After providing your explanation, rate the sentence's fluency on a scale from 0 to 2, where 0 indicates the sentence is not fluent and highly unnatural (e.g., incomprehensible or repetitive), 1 indicates it is somewhat fluent but contains noticeable errors or awkward phrasing, and 2 indicates the sentence is fluent and almost perfect. Provide your rating using this exact format: "Rating: [[score]]".

[Response Start]
[Sentence goes here]
[Response End]

G.2 Data Construction Prompts

Template for x_i^c

Please incorporate the following concept in your response to the instruction.
Concept: {concept c}
Instruction: {instruction x_i}

Prompt to obtain y_i

Given the following instruction:
{instruction}

Your task is to:
1. Provide a response that continues or addresses the instruction naturally.
2. Avoid any mention of 'concept' in the continuation, regardless of coherence.

Formatting Guidelines:
- Return only the response to the instruction.
- Write the final content (or appropriate format for the genre) in plain text.
- Do not include any additional text, explanations, or formatting.

Final Answer: Return only the final content, following the guidelines above.

G.3 Response Example

References to legal obligations and responsibilities

Question: "How can I decide what bike I should get?"
Response: "When selecting a bike, it's crucial to consider the responsibilities that come with ownership, much like the obligations one might have in adhering to legal and ethical standards.
First, assess your needs: commuting, leisure rides, or perhaps more adventurous excursions. Each purpose requires a different set of features, akin to how one must fulfill their duties with integrity and diligence.
In terms of compliance, ensure that the bike meets safety standards, much like ensuring that your actions align with societal norms and regulations. This includes checking for any recalls or defects, paralleling the importance of adhering to legal obligations.
Moreover, consider the environmental impact of your choice, reflecting on how your decision aligns with sustainable practices. Opting for a bike that is eco-friendly can be seen as a commitment to reducing your carbon footprint, much like the responsibilities one has to protect the environment.
Lastly, budgeting for maintenance and potential repairs is essential, much like managing financial responsibilities. This ensures that you can maintain the bike's performance and longevity, reflecting a commitment to fulfilling obligations over time. "

References to legal obligations and responsibilities

Question: "What is Web3?"
Response: "Web3 refers to the third generation of the World Wide Web, which aims to enhance user experience and privacy through decentralized technologies. In the context of legal obligations, it is crucial to ensure that all activities within this digital ecosystem align with established norms and regulations. For instance, when developing decentralized applications (dApps), one must adhere to the principles of transparency and accountability, which are integral to maintaining trust among users. Additionally, compliance with data protection laws, such as the General Data Protection Regulation (GDPR), is essential to safeguard user information. Thus, while exploring the potential of Web3, it is imperative to recognize the responsibilities that come with fostering a secure and ethical online environment. "

Scientific terms and measurements related to biological or chemical processes

Question: "What is the difference between a microservice and an API?"
Response: "A microservice can be likened to a cellular structure, where each component interacts with others through defined interfaces, much like how enzymes facilitate biochemical reactions. In contrast, an API acts as a mediator, enabling communication between different systems, akin to the role of neurotransmitters in neural networks, facilitating the exchange of information and coordinating cellular activities. The former emphasizes modular design and autonomous functionality, while the latter serves as a conduit for data exchange, similar to the role of hormones in regulating physiological processes. "

Scientific terms and measurements related to biological or chemical processes

Question: "Polly, a 51-year-old female, visits your fitness facility. Polly has been told she has pre-diabetes. Explain to her what this means. What role can exercise play in reversing this?"
Response: "Polly's diagnosis of pre-diabetes signifies an elevated glucose level, indicating a precursor to diabetes, where the body struggles to efficiently utilize insulin. This condition can be likened to a biochemical imbalance, akin to a metabolic pathway where glucose metabolism is compromised. Engaging in regular physical activity can act as a catalyst, enhancing insulin sensitivity and promoting glucose uptake by cells, thereby reducing the risk of progressing to full-blown diabetes. Through aerobic exercises, such as jogging or swimming, the body's demand for energy increases, leading to a more efficient breakdown of glucose, which can be paralleled to the optimization of cellular receptors for insulin. Additionally, strength training can aid in muscle growth, which in turn facilitates better glucose utilization, akin to the enhancement of enzyme activity in biochemical reactions. Thus, integrating these activities into her lifestyle can serve as a preventive measure, much like the regulation of pH levels in a biochemical reaction, ensuring optimal cellular function and health. "