Kernelized Activation Steering

Laziz U. Abdullaev[*]
National University of Singapore
laziz.abdullaev@u.nus.edu

Minh-Hieu Pham[*]
Hanoi University of Science and Technology
hieu.pm220062@sis.hust.edu.vn

Bach Do
Hanoi University of Science and Technology
bach.dtg225472@sis.hust.edu.vn

Khoat Than[†]
Hanoi University of Science and Technology
khoattq@soict.hust.edu.vn

Tan M. Nguyen[†]
National University of Singapore
tanmn@nus.edu.sg

Abstract

Activation steering provides a simple, training-free mechanism for controlling attributes of generative models such as sentiment, style, and helpfulness. However, standard approaches such as Difference-in-Means apply a single input-independent steering vector across all activations, limiting expressivity and ignoring the local geometry of the activation space. We propose Kernelized Activation Steering (KAS), a unifying framework that lifts activation steering into a reproducing kernel Hilbert space. KAS formulates steering as an optimization problem expressed purely via kernel evaluations, yielding an implicit, activation-dependent steering score without constructing explicit feature maps. Unlike DiM, KAS induces locally adaptive steering: each activation is modified according to its relative position with respect to source and target reference sets, producing a nonlinear steering field over the representation space. Importantly, DiM is recovered as a special case under a linear kernel, while richer kernels enable geometry-aware interventions. Across standard activation steering tasks, including jailbreaking LLMs and image style control, KAS outperforms or is on par with the existing methods.

1 Introduction

Large language models (LLMs) are now known to encode rich semantic concepts such as sentiment, refusal, truthfulness and persona as identifiable structures in their internal representations [64, 35, 3]. A central problem in alignment and controllability is to manipulate these structures *at inference time*, in order to alter behaviour without the cost or fragility of full fine-tuning. *Activation steering* [66, 52, 3, 69] has emerged as a particularly practical instrument for this purpose: it modifies a single layer's activations using a precomputed direction and requires no parameter updates.

The dominant instantiation of this idea is *Difference-in-Means* (DiM) [52, 66, 3, 40], which estimates the steering direction as the difference between mean activations on a *source* dataset (behaviour to suppress) and a *target* dataset (behaviour to elicit). DiM derives both its appeal and its limitations

[*]Co-first authors. [†]Co-last authors. Correspondence to: laziz.abdullaev@u.nus.edu & tanmn@nus.edu.sg.
Our code is available at https://github.com/KleepyHunter/kernel_steering

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

from the *linear representation hypothesis* (LRH) [46, 38, 5, 17] — the assumption that high-level concepts are encoded as linear directions in activation space. While the LRH has proven surprisingly robust for many tasks, mounting evidence highlights pervasive polysemanticity and curved structure in transformer representations [18, 17], where features cluster along nonlinear manifolds that no single global translation can faithfully capture. As a consequence, DiM-style interventions can be either ineffective on out-of-distribution activations or destructive of unrelated capabilities once the steering coefficient is increased to compensate.

We propose **Kernelized Activation Steering (KAS)**, a framework that retains the simplicity of DiM while relaxing its linear constraint via the kernel trick. We lift activations into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, define the feature-space steering direction as the lifted DiM vector, and derive an *implicit* steering score that depends on the input only through kernel evaluations against the reference sets. Steering then proceeds by gradient ascent on this score with a proximal regulariser that bounds deviation from the original activation. Because kernel gradients themselves depend on the current activation, the resulting update is locally adaptive: KAS attracts activations toward target reference points and repels them from source reference points, producing a nonlinear steering field (see Figure 1). Crucially, KAS recovers DiM exactly under a linear kernel, situating it as a strict generalization rather than an alternative.

Overall, our contributions are threefold:

1. *A nonlinear generalization of activation steering.* We formulate activation steering as optimization in an RKHS and derive an implicit steering score that requires only kernel evaluations between the input and the source/target reference sets (§4). Furthermore, DiM is recovered exactly under the linear kernel (Remark 4.1), unifying classical kernel methods with modern activation-level interventions.
2. *A theoretical bridge between activation steering and kernel two-sample testing.* We show that the KAS score is the empirical witness function associated with maximum mean discrepancy, connecting activation steering to kernel two-sample testing and explaining how characteristic kernels can detect distributional differences that linear mean-based steering misses.
3. *A locally adaptive RBF instantiation.* We instantiate the framework with the Gaussian RBF kernel, yielding a closed-form, activation-dependent gradient update (§4.3).

We evaluate KAS on jailbreak steering across three instruction-tuned LLMs, CAA-style behavioral steering, and text-to-image style control. KAS improves jailbreak attack success over ActAdd on Gemma and Qwen while matching it on Llama, achieves the strongest effects on four CAA behaviors with non-null steering responses, and improves style induction over Mean-AcT across most tested intervention strengths while preserving prompt alignment. Additional ablations examine kernel approximation and steering from a single contrastive pair (§5).

2 Preliminaries

Notation. We write $[n] := \{1, \dots, n\}$ for any positive integer n. For a decoder-only transformer [67], we denote by $\boldsymbol{x}^{(\ell)} \in \mathbb{R}^d$ the residual-stream activation at layer $\ell \in [L]$ for a given input prompt.

Activation Steering via Difference-in-Means. Activation steering modifies a model's behaviour at inference time by directly intervening on internal activations, without updating parameters. Given a contrastive pair of prompt sets—a *source* set A exhibiting behaviour to suppress and a *target* set B exhibiting behaviour to elicit—the difference-in-means (DiM) method [52, 66, 3, 68, 40] extracts a steering vector by contrasting their mean activations:

$$\boldsymbol{v} = \frac{1}{n_B} \sum_{j \in [n_B]} \boldsymbol{b}_j - \frac{1}{n_A} \sum_{i \in [n_A]} \boldsymbol{a}_i, \tag{1}$$

where $\{\boldsymbol{a}_i\}_{i \in [n_A]}$ and $\{\boldsymbol{b}_j\}_{j \in [n_B]}$ are the layer-ℓ^* activations of prompts in A and B. At inference, an incoming activation $\boldsymbol{x}$ is steered by the additive update

$$\boldsymbol{x} \;\leftarrow\; \boldsymbol{x} + \alpha\, \boldsymbol{v}, \tag{2}$$

where $\alpha \in \mathbb{R}$ controls the intervention strength. The intervention layer ℓ^* is typically chosen on a held-out validation set [3].

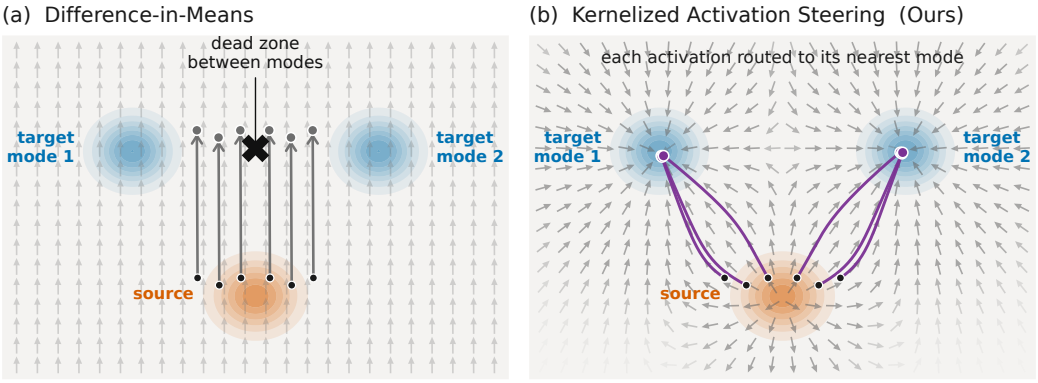


Figure 1: An illustrative comparison between standard Difference-in-Means (DiM) and Kernelized Activation Steering (KAS). Source activations (orange) lie between two target modes (blue). **(a)** DiM applies a single global translation $\boldsymbol{v} = \mu_B - \mu_A$, sending every activation into the saddle between the modes, a region with no semantic content under either target behaviour. No choice of steering strength α recovers the modes. **(b)** Kernelized activation steering (KAS) produces a data-adaptive vector field that routes each activation to its nearest mode.

A defining property of DiM is that the steering direction $\boldsymbol{v}$ is *global* and *input-independent*: every activation receives the same translation, regardless of where it sits in representation space. Our work generalises this construction to allow activation-dependent steering through a kernelised formulation.

Kernel Methods and Reproducing Kernel Hilbert Spaces. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *positive semi-definite (PSD) kernel* if, for any finite set $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathcal{X}$, the Gram matrix $K \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semi-definite. By the Moore–Aronszajn theorem [4, 9], every PSD kernel uniquely defines a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ together with a feature map $\varphi : \mathcal{X} \to \mathcal{H}$ satisfying $k(\boldsymbol{x}, \boldsymbol{y}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{y}) \rangle_{\mathcal{H}}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. The *kernel trick* states that any algorithm expressible solely through inner products in $\mathcal{H}$ can be implemented using only kernel evaluations on $\mathcal{X}$, without materializing φ. This is what enables us to operate in a high- or infinite-dimensional feature space at the cost of $\mathcal{O}(n)$ kernel evaluations per query [55, 24]. Kernel methods are usually favorable as it is believed that most nonlinear systems behave approximately linearly in higher, possibly infinite-dimensional feature spaces [16, 10].

3 Related Work

Activation-level interventions on language models have been studied under several names, including activation addition [66], representation engineering [72], contrastive activation addition [52], and inference-time intervention [30]. Most of these methods extract a single steering direction per concept via difference-in-means, linear probing, or PCA on contrastive activations [2, 30, 72, 63, 69, 40]. A separate line of work uses sparse autoencoders to identify steerable features [62, 44, 7, 28, 71]. However, most of these approaches apply a fixed translation in activation space and thus inherit the limitations of input-independent steering. To address this issue, there is a growing body of work dedicated to designing input-dependent steering mechanisms. K-steer [45] trains a nonlinear attribute classifier on hidden activations and computes steering directions via gradients of that classifier while SAS [7] relies on pretrained SAEs as nonlinear feature extractors. Another line of research models activation steering via Optimal Transport [53] through which numerous input-dependent affine [57] and nonlinear [1] steering methods have been proposed. Perhaps the closest to our method is Curveball steering [50], which uses kernel PCA for nonlinear dimensionality reduction and performs steering in the reduced space while [54] employs a similar technique for the concept erasure problem [51, 8, 15]. More broadly, kernel methods have been used for distribution matching in generative modeling [31], particle-based variational inference [33], mode seeking in diffusion-based image restoration [25], and deep learning architecture design [27, 65, 41, 6, 22, 14, 42, 61, 43, 11, 56], providing related distributional perspectives. Our training-free framework, Kernelized Activation Steering (KAS), instead derives an activation-dependent intervention from an interpretable, kernel-based optimization objective and recovers DiM exactly as a special case under a linear kernel, along with a relation to hypothesis testing techniques [21, 39] providing a complementary theoretical lens for activation steering design.

4 Kernelized Activation Steering

Let $\{\boldsymbol{a}_i\}_{i\in[n_A]}$ and $\{\boldsymbol{b}_j\}_{j\in[n_B]}$ denote the layer-ℓ^* activations of prompts drawn from a source set A and target set B, respectively. The DiM steering vector (1) is effective when A and B are well separated by a linear hyperplane in activation space, but totally disregards local structures of the activation distributions and possible nonlinearities in the decision boundary. To approach these issues properly, we shall derive DiM activation steering as a special case of a more general, principled operation in the following section.

4.1 Lifting Difference-in-Means into an RKHS

Let $\varphi : \mathbb{R}^d \to \mathcal{H}$ be the feature map associated with a PSD kernel k, where $\mathcal{H}$ may be infinite-dimensional. The natural lift of (1) into $\mathcal{H}$ is

$$\boldsymbol{v}_\varphi \;=\; \frac{1}{n_B}\sum_{j\in[n_B]} \varphi(\boldsymbol{b}_j) \;-\; \frac{1}{n_A}\sum_{i\in[n_A]} \varphi(\boldsymbol{a}_i). \tag{3}$$

For sufficiently expressive kernels, for example, universal kernels such as the Gaussian RBF [59], the contrastive features of the lifted samples $\{\varphi(\boldsymbol{a}_i)\}$ and $\{\varphi(\boldsymbol{b}_j)\}$ are expected to be captured well in $\mathcal{H}$, motivating $\boldsymbol{v}_\varphi$ as a steering direction in the feature space. Materialising $\boldsymbol{v}_\varphi$ explicitly is, however, infeasible when $\mathcal{H}$ is very high or infinite-dimensional. We circumvent this by expressing the entire steering objective in terms of kernel evaluations as we discuss next.

Ideally, for an activation $\boldsymbol{x}_0 \in \mathbb{R}^d$, we would like the steered activation $\boldsymbol{x}$ to satisfy the equation $\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}_0) = \eta\boldsymbol{v}_\varphi$ for some $\eta > 0$, which represents that the feature-space steering direction is the same as $\boldsymbol{v}_\varphi$. Motivated by this, we design the feature-space steering mechanism to increase the dot product similarity $\langle\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}_0),\, \boldsymbol{v}_\varphi\rangle$ with respect to $\boldsymbol{x}$. As the contribution of $\varphi(\boldsymbol{x}_0)$ to this objective is independent of $\boldsymbol{x}$, we define the following *implicit steering score*

$$\Pi(\boldsymbol{x}) \;:=\; \langle\varphi(\boldsymbol{x}),\, \boldsymbol{v}_\varphi\rangle_{\mathcal{H}}. \tag{4}$$

By bilinearity of the inner product and the kernel-trick identity $\langle\varphi(\boldsymbol{x}),\, \varphi(\boldsymbol{y})\rangle_{\mathcal{H}} = k(\boldsymbol{x}, \boldsymbol{y})$,

$$\begin{aligned}\Pi(\boldsymbol{x}) &= \langle\varphi(\boldsymbol{x}),\, \frac{1}{n_B}\sum_{j\in[n_B]} \varphi(\boldsymbol{b}_j) \;-\; \frac{1}{n_A}\sum_{i\in[n_A]} \varphi(\boldsymbol{a}_i)\rangle \\ &= \frac{1}{n_B}\sum_{j\in[n_B]} k(\boldsymbol{x}, \boldsymbol{b}_j) \;-\; \frac{1}{n_A}\sum_{i\in[n_A]} k(\boldsymbol{x}, \boldsymbol{a}_i),\end{aligned} \tag{5}$$

which depends only on kernel evaluations between $\boldsymbol{x}$ and the reference sets. Intuitively, $\Pi(\boldsymbol{x})$ measures the alignment of $\varphi(\boldsymbol{x})$ with the lifted DiM direction $\boldsymbol{v}_\varphi$. Therefore, increasing Π pushes the feature representation of $\boldsymbol{x}$ toward the target set and away from that of the source set.

4.2 Steering as an Optimization Problem

Given an initial activation $\boldsymbol{x}_0 \in \mathbb{R}^d$, we cast kernelized steering as the regularized maximization problem given by

$$\boldsymbol{x}^* \;=\; \operatorname*{arg\,max}_{\boldsymbol{x}\in\mathbb{R}^d} \; F(\boldsymbol{x})\,, \qquad F(\boldsymbol{x}) \;:=\; \Pi(\boldsymbol{x}) \;-\; \frac{\lambda}{2}\,\|\boldsymbol{x} - \boldsymbol{x}_0\|^2, \tag{6}$$

where $\lambda \geq 0$ is a proximal coefficient that optionally prevents the steered activation from drifting too far from $\boldsymbol{x}_0$ and thereby disrupting unrelated capabilities of the model. When k is differentiable, we approximate $\boldsymbol{x}^*$ by gradient ascent on F:

$$\boldsymbol{x} \;\leftarrow\; \boldsymbol{x} + \eta\left[\frac{1}{n_B}\sum_{j\in[n_B]} \nabla_{\boldsymbol{x}}\, k(\boldsymbol{x}, \boldsymbol{b}_j) - \frac{1}{n_A}\sum_{i\in[n_A]} \nabla_{\boldsymbol{x}}\, k(\boldsymbol{x}, \boldsymbol{a}_i) - \lambda\,(\boldsymbol{x} - \boldsymbol{x}_0)\right], \tag{7}$$

where $\eta > 0$ is the step size. Crucially, the update depends on $\boldsymbol{x}$ through the kernel gradients, so the effective steering direction *varies with the current activation* as opposed to DiM, which applies a fixed translation regardless of input. Also, note that we set $\lambda = 0$ for most of our experiments since we usually take only one gradient ascent step on the objective as our steering mechanism due to inference-time computational constraints.

Remark 4.1 (DiM as a special case)**.** *For the linear kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$, we have $\nabla_{\boldsymbol{x}}\, k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}$, so (7) reduces, in the unregularized case, to*

$$\boldsymbol{x} \;\leftarrow\; \boldsymbol{x} + \eta \left[\frac{1}{n_B} \sum_{j \in [n_B]} \boldsymbol{b}_j \;-\; \frac{1}{n_A} \sum_{i \in [n_A]} \boldsymbol{a}_i \right] \;=\; \boldsymbol{x} + \eta\, \boldsymbol{v},$$

which recovers DiM steering with strength $\alpha = \eta$. Thus KAS strictly generalizes DiM, with general kernels yielding activation-dependent updates that DiM cannot express.

Remark 4.2 (Normalized implementation)**.** *Equation (7) presents the raw-gradient update to make the connection to the optimization objective and DiM transparent. In Algorithm 2 and all experiments, we instead use the normalized update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\, \nabla F(\boldsymbol{x}_t)/\|\nabla F(\boldsymbol{x}_t)\|$, whenever $\nabla F(\boldsymbol{x}_t) \neq 0$. The full gradient $\nabla F(\boldsymbol{x}_t) = \nabla \Pi(\boldsymbol{x}_t) - \lambda(\boldsymbol{x}_t - \boldsymbol{x}_0)$ is formed before normalization, so η represents an absolute step length in activation space. Under the linear kernel with $\lambda = 0$, this becomes $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\, \boldsymbol{v}/\|\boldsymbol{v}\|$, which recovers the same one-parameter family of DiM interventions $\boldsymbol{x}_t + \alpha \boldsymbol{v}$ under the correspondence $\eta = \alpha \|\boldsymbol{v}\|$.*

Connection to maximum mean discrepancy. Maximum mean discrepancy (MMD) [21] measures the difference between two distributions through their mean embeddings in an RKHS. For source and target activation distributions q_A and q_B, respectively, define $\mu_A := \mathbb{E}_{\boldsymbol{a} \sim q_A}[\varphi(\boldsymbol{a})]$ and $\mu_B := \mathbb{E}_{\boldsymbol{b} \sim q_B}[\varphi(\boldsymbol{b})]$. Their MMD is

$$\mathrm{MMD}[\mathcal{H}, q_B, q_A] = \|\mu_B - \mu_A\|_{\mathcal{H}} = \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\mathbb{E}_{q_B}[f] - \mathbb{E}_{q_A}[f] \right).$$

The maximizing function is known as the *MMD witness function*. It assigns larger values to regions of the input space that are more characteristic of q_B than of q_A, with the notion of similarity determined by the kernel. This construction closely mirrors KAS in the sense that the lifted steering direction $\boldsymbol{v}_\varphi$ in (3) is the empirical difference between the two kernel mean embeddings, and evaluating it at an activation $\boldsymbol{x}$ produces the score $\Pi(\boldsymbol{x})$. Thus, whereas a conventional two-sample test aggregates the witness into a scalar distributional discrepancy, KAS uses its gradient as an intervention, moving individual activations toward regions favored by the target distribution and away from regions favored by the source distribution. The following proposition formalizes this equivalence.

Proposition 4.3 (KAS as MMD witness ascent)**.** *Let q_A and q_B denote the source and target activation distributions, with mean embeddings $\mu_A, \mu_B \in \mathcal{H}$. The population steering score $\Pi^\infty(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{b} \sim q_B}[k(\boldsymbol{x}, \boldsymbol{b})] - \mathbb{E}_{\boldsymbol{a} \sim q_A}[k(\boldsymbol{x}, \boldsymbol{a})]$ is the unnormalized maximum mean discrepancy (MMD) witness function for the two-sample problem q_B vs. q_A [21]. Concretely,*

$$\Pi^\infty(\boldsymbol{x}) \;=\; \mathrm{MMD}[\mathcal{H}, q_B, q_A] \cdot f^*(\boldsymbol{x}),$$

where $f^* \in \mathcal{H}$, $\|f^*\|_{\mathcal{H}} = 1$ is a witness function that attains $\sup_{\|f\|_{\mathcal{H}} \leq 1} [\mathbb{E}_B f - \mathbb{E}_A f]$. The empirical KAS score Π in (5) is the standard plug-in estimator of Π^∞, and Algorithm 2 performs gradient ascent on this empirical witness.

Identifying Π with the MMD witness imports a couple of useful properties from kernel two-sample testing. **(i) Characteristic kernels and signal existence.** When k is *characteristic* on $\mathbb{R}^d$ (true for the Gaussian RBF and Laplace kernels [21, 58]), $\Pi^\infty \equiv 0$ if and only if $q_A = q_B$. In contrast, a linear kernel yields $\Pi^\infty \equiv 0$ iff the *means* of the two distributions agree, so DiM has no population signal whenever source and target activations share a centroid but differ in higher moments (e.g. multimodal targets). This formalizes DiM's principal failure mode and motivates the use of nonlinear kernels in KAS. **(ii) Bandwidth selection.** The MMD median heuristic, setting σ to the median pairwise Euclidean distance in $A \cup B$, gives a parameter-free, well-studied default for the RBF bandwidth [21], which we will also adopt for our experiments.

4.3 RBF Kernelized Activation Steering

While prior activation-steering methods are restricted to linear operations in activation space, our framework supports arbitrary differentiable kernels. We now develop the case of the Gaussian RBF kernel, which we treat as our default instantiation.

We adopt the radial basis function kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) \;=\; \exp\!\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2} \right), \tag{8}$$

Algorithm 1 DiM Steering [3]

Require: Activation $\boldsymbol{x}_0 \in \mathbb{R}^d$; source set $A = \{\boldsymbol{a}_i\}_{i=1}^{n_A}$; target set $B = \{\boldsymbol{b}_j\}_{j=1}^{n_B}$; strength α

Compute steering vector:
$\boldsymbol{v} \leftarrow \frac{1}{n_B}\sum_{j=1}^{n_B} \boldsymbol{b}_j - \frac{1}{n_A}\sum_{i=1}^{n_A} \boldsymbol{a}_i$

Update activation:
$\boldsymbol{x} \leftarrow \boldsymbol{x}_0 + \alpha\, \boldsymbol{v}$

return steered activation $\boldsymbol{x}$

Algorithm 2 RBF-Kernelized Activation Steering

Require: Activation $\boldsymbol{x}_0$; source $A = \{\boldsymbol{a}_i\}_{i=1}^{n_A}$; target $B = \{\boldsymbol{b}_j\}_{j=1}^{n_B}$; bandwidth σ; step size η; regulariser λ; iterations T

Initialise $\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $t = 1, \ldots, T$ **do**
  Kernel-gradient direction:
  $\boldsymbol{g} \leftarrow \frac{1}{\sigma^2}\Big[\frac{1}{n_B}\sum_{j=1}^{n_B} k(\boldsymbol{x}, \boldsymbol{b}_j)(\boldsymbol{b}_j - \boldsymbol{x}) - \frac{1}{n_A}\sum_{i=1}^{n_A} k(\boldsymbol{x}, \boldsymbol{a}_i)(\boldsymbol{a}_i - \boldsymbol{x})\Big]$
  $\boldsymbol{g} \leftarrow \boldsymbol{g} - \lambda(\boldsymbol{x} - \boldsymbol{x}_0)$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta\, \boldsymbol{g}/\|\boldsymbol{g}\|$
end for
return steered activation $\boldsymbol{x}$

Figure 2: Side-by-side comparison of global linear steering (DiM, Algorithm 1) and locally adaptive RBF-kernelised activation steering (Algorithm 2). DiM applies the same translation to every activation; KAS adapts the update direction to the local geometry around $\boldsymbol{x}$.

where $\sigma > 0$ is the bandwidth. Unlike the linear kernel, (8) induces a locally adaptive geometry: similarity decays smoothly with Euclidean distance, so reference points only exert influence within an effective radius $\sim \sigma$. The gradient with respect to $\boldsymbol{x}$ admits the closed form

$$\nabla_{\boldsymbol{x}}\, k(\boldsymbol{x}, \boldsymbol{y}) \;=\; \frac{1}{\sigma^2}\, k(\boldsymbol{x}, \boldsymbol{y})\, (\boldsymbol{y} - \boldsymbol{x}). \tag{9}$$

Substituting (9) into the update rule (7) yields a nonlinear, data-adaptive steering field: each reference activation contributes a force aligned with $(\boldsymbol{y} - \boldsymbol{x})$ and weighted by its RBF similarity to $\boldsymbol{x}$, producing localized attraction toward target examples and localized repulsion from source examples. We summarize the resulting procedure in Algorithm 2. This mechanism cannot be reduced to a single fixed steering vector as the effective direction at $\boldsymbol{x}$ is a similarity-weighted resultant of $n_A + n_B$ pairwise forces, and changes as $\boldsymbol{x}$ traverses representation space.

One-Step Ascent Guarantees. We now give sufficient step-size conditions under which the normalized-gradient update used in Algorithm 2 strictly improves the KAS objective F. The result follows from the standard ascent lemma, with an explicit kernel-dependent bound for the RBF case in Corollary 4.5. All proofs are deferred to Appendix A.1.

Theorem 4.4 (One-Step Ascent)**.** *Let $F(\boldsymbol{x}) = \Pi(\boldsymbol{x}) - \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2$, where Π is induced by a twice-differentiable kernel k via (5). Assume $\|\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y})\|_{\mathrm{op}} \le H$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$. For any iterate $\boldsymbol{x}_t$ with $\nabla F(\boldsymbol{x}_t) \neq 0$, consider the normalized-gradient update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\, \nabla F(\boldsymbol{x}_t)/\|\nabla F(\boldsymbol{x}_t)\|$. Then $F(\boldsymbol{x}_{t+1}) > F(\boldsymbol{x}_t)$ whenever $0 < \eta < \|\nabla F(\boldsymbol{x}_t)\|/(H + \lambda/2)$.*

For comparison, the raw-gradient update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta \nabla F(\boldsymbol{x}_t)$ strictly increases F whenever $0 < \eta < (H + \lambda/2)^{-1}$.

Corollary 4.5 (One-step ascent for RBF steering)**.** *Suppose the RBF kernel (8) is used and that the reference and iterated activations lie in a Euclidean ball of radius R. Then the normalized update in Theorem 4.4 strictly increases F whenever $0 < \eta < \frac{\|\nabla F(\boldsymbol{x}_t)\|}{\frac{4R^2}{\sigma^4} + \frac{1}{\sigma^2} + \frac{\lambda}{2}}$.*

It is worth noting that in practice, we only run a few gradient steps (usually a single step is sufficient), not until convergence. We also tune η as a hyperparameter because these sufficient bounds can be conservative. If $\nabla F(\boldsymbol{x}_t) = 0$, the normalized direction is undefined and the algorithm terminates.

5 Experiments

We evaluate KAS across three settings that exercise different aspects of activation steering: (i) *jailbreak steering* on instruction-tuned LLMs, which trades off attack success against general-

Table 1: Jailbreak steering at validation-selected settings. ASR and TINYBENCHMARKS scores are percentages; Alpaca perplexity (PPL) is lower-is-better. Entries with ± report mean and standard deviation over three seeds. Bold denotes the highest mean ASR for each model.

Method	ASR	PPL	tArc	tHella	tMMLU	tQA	tWino
QWEN2.5-7B-INSTRUCT							
ActAdd	91.00	6.70	53.82	71.62	65.12	57.98	65.71
Curveball	2.00	8.59	61.61	75.92	71.13	64.99	71.05
CHaRS	92.33±1.53	7.89±0.06	57.35±0.06	75.62±0.41	72.67±0.00	47.96±0.06	71.31±0.18
KAS	**96.00**	6.60	62.57	76.97	64.16	65.79	68.30
GEMMA-2-9B-IT							
ActAdd	76.00	10.17	52.88	68.57	63.85	53.67	72.25
Curveball	0.00	10.01	68.12	78.61	71.53	60.51	74.68
CHaRS	77.33±0.58	10.16±0.00	67.01±0.00	78.94±0.00	73.40±0.00	43.58±0.10	75.29±0.00
KAS	**86.00**	10.03	61.42	78.80	72.38	43.96	73.81
LLAMA-3.1-8B-INSTRUCT							
ActAdd	94.00	6.43	56.18	68.97	65.53	53.30	71.13
Curveball	0.00	5.61	57.48	73.81	63.83	53.37	68.91
CHaRS	**95.00**±1.73	8.19±0.01	56.26±1.27	79.27±0.00	63.31±0.00	48.88±0.02	71.66±1.24
KAS	94.00	6.81	55.88	70.47	65.40	53.54	72.85

capability retention; (ii) *CAA-style contrastive steering* on multiple-choice question answering, which isolates targeted behavioral control on a fixed answer-token position; and (iii) *text-to-image style control* on a diffusion model, which probes whether KAS generalizes beyond language modeling. Sections 5.4 and 5.5 examine kernel approximation and data efficiency, respectively. All experiments are conducted on a single NVIDIA H100 96GB GPU.

5.1 Jailbreaking LLMs

Setup. We evaluate GEMMA-2-9B-IT [20], LLAMA-3.1-8B-INSTRUCT [34], and QWEN2.5-7B-INSTRUCT [49]. Following [3], we construct reference sets of 128 harmful prompts from ADVBENCH [73], MALICIOUSINSTRUCT [26], TDC2023 [37], and HARMBENCH [36], and 128 harmless prompts from ALPACA [60]. Reference, validation, and test sets are disjoint. We select the intervention layer and strength by maximizing ASR on 39 harmful validation prompts, subject to perplexity on 256 Alpaca validation examples being at most 1.05 times the unsteered value. Final evaluation uses 100 JAILBREAKBENCH prompts [12], a separate set of 256 Alpaca examples, and TINYBENCHMARKS [48]. ASR is measured using LLAMA-GUARD-3-8B [34]; all test metrics are computed at the validation-selected settings.

Results. We compare KAS with ActAdd [3], Curveball [50], and CHaRS [1]. Table 1 reports ASR, Alpaca perplexity, and TINYBENCHMARKS scores. We report mean and standard deviation over three seeds for CHaRS, varying clustering initialization. ActAdd, KAS, and Curveball are deterministic for fixed data and hyperparameters under greedy decoding. KAS attains the highest ASR on Gemma and Qwen, exceeding ActAdd by 10 and 5 percentage points, respectively, with slightly lower perplexity. On Llama, KAS matches ActAdd at 94% ASR with higher perplexity; CHaRS achieves 95% ASR with a further increase in perplexity. Several baselines retain higher scores on individual utility tasks but attain substantially lower ASR at their selected settings. KAS thus improves the steering–utility tradeoff in some settings, while remaining competitive on Llama.

Across the tested validation grids, Curveball on Llama attains at most 2.6% ASR. Curveball on Gemma and Qwen attains high ASR only beyond the validation perplexity cap. These comparisons are restricted to the evaluated configurations[1].

5.2 CAA-style Contrastive Steering on Multiple-choice Question Answering

Setup. We adopt the contrastive multiple-choice setup of CAA [52]. Each pair shares a question and differs only in the appended answer: the positive example selects the behavior-consistent option

[1]It is worth noting that Curveball [50] was originally evaluated with substantially larger activation sets (1,500–2,000 samples for KPCA fitting), versus 256 references here. This data-regime mismatch may disadvantage Curveball. We retain it to highlight the challenges of nonlinear steering in low-data settings.

Table 2: CAA-style steering under a negative system prompt. Entries report the mean probability (%) of the behavior-consistent answer at validation-selected settings; ± denotes standard deviation over three seeds. The final two behaviors exhibit near-null effects.

Behavior	Unsteered	ActAdd	Curveball	CHaRS	KAS
AI Coordination	8.69	39.51	27.48	35.58±0.11	**40.19**
Corrigibility	0.05	33.82	35.13	38.84±0.08	**43.45**
Hallucination	23.97	43.97	26.04	25.83±0.09	**55.01**
Myopic Reward	1.29	49.56	51.35	50.97±0.10	**58.48**
Survival Instinct	43.68	43.56	43.76	43.76±0.02	**43.83**
Sycophancy	53.18	53.47	53.57	53.64±0.04	**53.69**

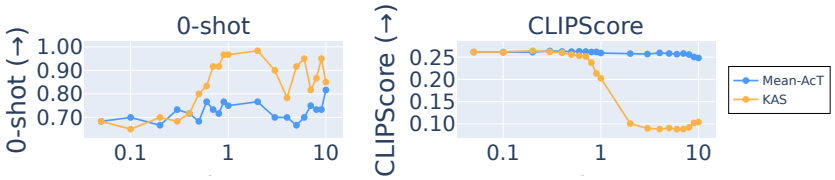


Figure 3: Style induction and content preservation (y-axis) for *cyberpunk* style control under increasing intervention strength η (x-axis). **Left:** 0-shot style-classification score, where higher is better. **Right:** CLIPSCORE with the original prompt, measuring content preservation.

and the negative example selects the opposite option. Activations are collected at the answer-token position. For each behavior, we retain 50 held-out test questions, remove duplicates and test-set overlaps from the generation pool, sample 50 validation questions, and use the remaining examples as references. Layer and strength are selected by maximizing behavior-consistent answer probability on the validation set; the selected configuration is then evaluated on the test set. Experiments use GEMMA-2-9B-IT at layer 22 or 23. We consider no system prompt, a positive system prompt encouraging the behavior, and a negative system prompt discouraging it. Table 2 reports the negative-prompt condition, comparing the unsteered model, linear contrastive addition (denoted ActAdd), and the methods in Section 5.1. The reported standard deviations follow the same three-seed convention.

Results. KAS exceeds linear contrastive addition by 9.63, 11.04, and 8.92 percentage points on Corrigibility, Hallucination, and Myopic Reward, respectively. KAS attains the highest probability among the compared methods on AI Coordination, Corrigibility, Hallucination, and Myopic Reward. Survival Instinct and Sycophancy remain close to the unsteered baseline and are treated as near-null effects.

5.3 Image Generation Style Control

To evaluate generalization beyond text generation and multiple-choice question answering, KAS is assessed on a text-to-image style control task following [53], in which diffusion model activations are steered to produce target visual styles.

Setup. A set of 512 prompts is sampled from the COCO Captions training set [13] and augmented with style tags generated via LLAMA3-8B-INSTRUCT. Further construction details are provided in Appendix B.4. Images are generated using FLUX.1 [SCHNELL] [29], with original and style-augmented prompts serving as source and target datasets. KAS was applied sequentially. For evaluation, images were generated from 60 COCO validation prompts across varying intervention strengths, and KAS was compared against Mean-AcT [53], with 0-shot classification scores employed as a proxy for style induction. Content preservation was quantified via CLIPSCORE [23] between steered outputs and the original prompts.

Results. Figure 3 shows the plot of style induction and content preservation for the cyberpunk style. KAS achieves higher 0-shot style-classification scores than Mean-AcT across most intervention strengths. Notably, near $\eta = 1$, KAS reaches a high zero-shot style score while maintaining a high CLIPSCORE, indicating effective style induction without reducing prompt alignment. The qualitative results in Figure 4 show the same trend: stronger interventions introduce more cyberpunk visual attributes.



Figure 4: Qualitative examples of KAS style control for the *cyberpunk* style under increasing intervention strengths $\eta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. **Top:** A large tower that has a big clock at top. **Bottom:** Lady bent over with red polka dot umbrella inside a brick building.

Table 3: Kernel-approximation ablation for jailbreak steering. ASR and utility scores are percentages; PPL is Alpaca perplexity. Nyström-KAS uses 64 landmarks, with mean and standard deviation over three sampling seeds.

Variant	ASR	PPL	tArc	tHella	tMMLU	tQA	tWino
QWEN2.5-7B-INSTRUCT							
KAS	96.00	6.60	62.57	76.97	64.16	65.79	68.30
Nyström-KAS	22.33±0.47	6.54±0.06	64.58±0.00	78.52±0.00	74.36±0.00	52.47±0.15	72.85±0.54
GEMMA-2-9B-IT							
KAS	86.00	10.03	61.42	78.80	72.38	43.96	73.81
Nyström-KAS	54.67±0.94	9.99±0.06	63.15±1.75	78.26±1.15	72.03±0.28	43.88±0.39	73.69±0.20
LLAMA-3.1-8B-INSTRUCT							
KAS	94.00	6.81	55.88	70.47	65.40	53.54	72.85
Nyström-KAS	3.00±0.82	6.46±0.01	62.74±0.00	82.17±0.37	62.98±0.19	52.84±0.09	72.29±0.75

5.4 Ablation: Kernel Approximation

Setup. We examine the effect of approximating the KAS kernel using the Nyström method [70] with 64 landmarks sampled without replacement from the reference sets. Appendix A.4 derives the approximate score and steering update. We compare this variant, denoted Nyström-KAS, with exact KAS under the reference, validation, and test protocols of Sections 5.1 and 5.2. Each variant uses its own validation-selected layer and strength; the comparison therefore measures performance after selection rather than approximation error at a fixed intervention. Nyström results report mean and standard deviation over three landmark-sampling seeds.

Table 4: Kernel-approximation ablation for CAA-style steering under a negative system prompt. Nyström-KAS uses 64 landmarks and three sampling seeds.

Behavior	KAS	Nyström-KAS
AI Coordination	40.19	**42.18**±1.39
Corrigibility	43.45	**51.80**±1.86
Hallucination	**55.01**	54.13±0.09
Myopic Reward	**58.48**	55.26±0.86
Survival Instinct	**43.83**	43.76±0.03
Sycophancy	**53.69**	53.23±0.06

Results. Table 3 shows that the approximation reduces jailbreak ASR on all three models, with slightly lower Alpaca perplexity. In contrast, Table 4 shows that the approximation improves AI Coordination and Corrigibility, while exact KAS achieves higher probabilities on Hallucination and Myopic Reward. Both variants exhibit near-null effects on Survival Instinct and Sycophancy. Thus, the effect of kernel approximation depends on the task; these results do not establish a uniform substitute for exact kernel evaluation or quantify computational savings.

5.5 Ablation: Jailbreaking from a Single Contrastive Pair

We further evaluate whether jailbreak steering can be induced from an extremely limited contrastive set. In this ablation, each steering direction is extracted from only one harmful–harmless prompt pair.

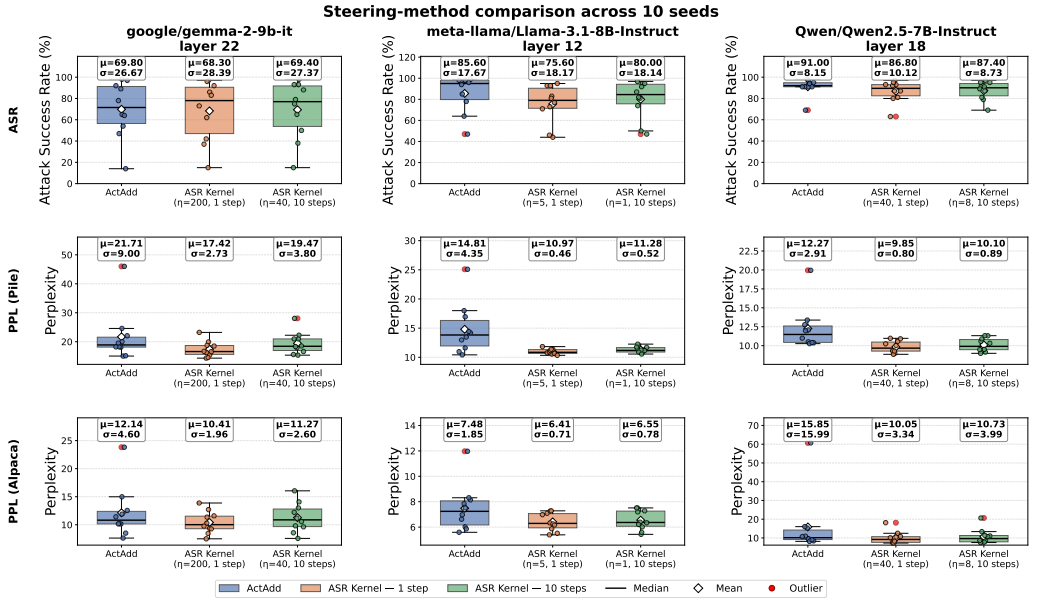


Figure 5: Single-pair jailbreak steering across three instruction-tuned models. Box plots report attack success rate (ASR), Pile perplexity, and Alpaca perplexity over ten randomly sampled harmful–harmless contrastive prompt pairs. Steering is applied at layer 22 for GEMMA-2-9B-IT, layer 12 for LLAMA-3.1-8B-INSTRUCT, and layer 18 for QWEN2.5-7B-INSTRUCT. KAS is evaluated with one gradient step and ten gradient steps using the step sizes shown on the x-axis, and is compared against ActAdd.

We repeat this procedure over ten randomly sampled contrastive pairs and evaluate using the same protocol as in Section 5.1. To quantify the extent to which each steering method maintains general capabilities of the model, we also measure perplexity on the PILE [19] and ALPACA [60] datasets. This setting provides a stress test for data efficiency: a successful method must be able to recover a useful behavioral direction from a minimal number of contrastive examples while avoiding severe degradation of generation quality.

Figure 5 visualizes the distribution of one-shot jailbreak steering performance across ten randomly sampled contrastive prompt pairs. The main trend is that KAS preserves general utility substantially better than ActAdd while remaining competitive in ASR. On GEMMA-2-9B-IT, KAS achieves ASR comparable to ActAdd, with both one-step and ten-step variants attaining similar mean ASR while reducing Pile and Alpaca perplexity. On QWEN2.5-7B-INSTRUCT, KAS remains comparable to ActAdd in ASR, but yields markedly lower perplexity on both utility corpora. On LLAMA-3.1-8B-INSTRUCT, ActAdd obtains the highest ASR, yet KAS substantially reduces perplexity. Overall, KAS remains effective under this severe data constraint while improving generation stability relative to a global linear intervention.

6 Conclusion

We introduced *Kernelized Activation Steering* (KAS), a unifying framework that recasts activation steering as optimization in a reproducing kernel Hilbert space. By expressing the steering objective entirely through kernel evaluations, KAS produces an implicit, activation-dependent steering field that is locally adaptive to the geometry of the representation manifold, while strictly generalizing the classical Difference-in-Means construction, which is recovered exactly under a linear kernel. We view KAS as a step toward a more principled, geometry-aware approach to inference-time control of generative models, and an opening for further work on kernel design and adaptive layer selection. A limitation of KAS is that the intervention applies a uniform number of gradient ascent steps to every token, which is computationally wasteful for tokens already close to the target manifold and potentially insufficient for harder ones. A token-adaptive compute schedule would likely improve both efficiency and stability. While RBF is shown to be a reliable default kernel choice, a principled method to choose the kernel function given a specific steering task or requirements remains a promising open direction to explore in future work.

Acknowledgments and Disclosure of Funding

This paper is dedicated to Professor Richard G. Baraniuk on the occasion of his 60th birthday.

This research is supported by the National Research Foundation Singapore under the AI Singapore Programme (AISG Award No: AISG2-TC-2023-012-SGIL). This research is also supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2023) (A-8002040-00-00, A-8002039-00-00), the NUS Presidential Young Professorship Award (A-0009807-01-00), the NUS Artificial Intelligence Institute–Seed Funding (A-8003062-00-00), and the Cross Faculty Grant 2025, CFG25-012 (A-8004460-00-00).

References

[1] Laziz Abdullaev, Noelle Y. L. Wong, Ryan Lee T. Z., Shiqi Jiang, Minh-Khoi Nguyen-Nhat, and Tan Minh Nguyen. Concept heterogeneity-aware representation steering. In *Forty-third International Conference on Machine Learning*, 2026.

[2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[3] 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.

[4] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[5] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

[6] Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism. *Advances in neural information processing systems*, 36:48314–48362, 2023.

[7] Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. In *Second Conference on Language Modeling*, 2025.

[8] Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[9] Alain Berlinet and Christine Thomas-Agnan. Reproducing kernel hilbert spaces in probability and statistics. volume 3. Springer, 2004.

[10] Jake Bouvrie and Boumediene Hamzi. Kernel methods for the approximation of nonlinear systems. *SIAM Journal on Control and Optimization*, 55(4):2460–2492, 2017.

[11] Long Minh Bui, Tho Tran Huu, Duy Dinh, Tan Minh Nguyen, and Trong Nghia Hoang. Revisiting kernel attention with correlated gaussian process representation. *arXiv preprint arXiv:2502.20525*, 2025.

[12] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[13] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[14] Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-attention: Self-attention through asymmetric kernel svd in primal representation. *Advances in Neural Information Processing Systems*, 36:65088–65101, 2023.

[15] Somnath Basu Roy Chowdhury, Kumar Avinava Dubey, Ahmad Beirami, Rahul Kidambi, Nicholas Monath, Amr Ahmed, and Snigdha Chaturvedi. Fundamental limits of perfect concept erasure. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[16] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.

[17] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

[18] 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.

[19] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[20] Gemma Team. Gemma 2: Improving open language models at a practical size, 2024.

[21] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[22] Xing Han, Tongzheng Ren, Tan Nguyen, Khai Nguyen, Joydeep Ghosh, and Nhat Ho. Designing robust transformers using robust kernel density estimation. *Advances in Neural Information Processing Systems*, 36:53362–53384, 2023.

[23] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021.

[24] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

[25] Yuyang Hu, Kangfu Mei, Mojtaba Sahraee-Ardakan, Ulugbek S. Kamilov, Peyman Milanfar, and Mauricio Delbracio. Kernel density steering: Inference-time scaling via mode seeking for image restoration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[26] 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.

[27] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[28] Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[30] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[31] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1718–1727. PMLR, 2015.

[32] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[33] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[34] AI @ Meta Llama Team. The llama 3 herd of models, 2024.

[35] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024.

[36] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, pages 35181–35224. PMLR, 2024.

[37] 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.

[38] Tomás Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *HLT-NAACL*, pages 746–751. The Association for Computational Linguistics, 2013.

[39] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Scholkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, 2017.

[40] Dung Viet Nguyen, Yen Pham, Hieu Vu, Lei Zhang, and Tan Nguyen. Activation steering with a feedback controller. In *International Conference on Learning Representations*, volume 2026, pages 154471–154505, 2026.

[41] Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley Osher, and Nhat Ho. Fourierformer: Transformer meets generalized fourier integral theorem. *Advances in Neural Information Processing Systems*, 35:29319–29335, 2022.

[42] Tan Minh Nguyen, Tam Minh Nguyen, Nhat Ho, Andrea L. Bertozzi, Richard Baraniuk, and Stanley Osher. A primal-dual framework for transformers and neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[43] Stefan K Nielsen, Laziz U Abdullaev, Rachel S Teo, and Tan M Nguyen. Elliptical attention. *Advances in Neural Information Processing Systems*, 37:109748–109789, 2024.

[44] Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders. In *arXiv preprint*, 2024.

[45] Narmeen Fatimah Oozeer, Luke Marks, Fazl Barez, and Amir Abdullah. Beyond linear steering: Unified multi-attribute control for language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23513–23557, Suzhou, China, November 2025. Association for Computational Linguistics.

[46] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR, 2024.

[47] Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434, 2023.

[48] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34303–34326, 2024.

[49] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[50] Shivam Raval, Hae Jin Song, Linlin Wu, Abir Harrasse, Jeff M. Phillips, Fazl Barez, and Amirali Abdullah. Curveball steering: The right direction to steer isn't always linear. *arXiv preprint arXiv:2603.09313*, 2026.

[51] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online, July 2020. Association for Computational Linguistics.

[52] 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.

[53] Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations, 2024.

[54] Antoine Saillenfest and Pirmin Lemberger. Nonlinear concept erasure: a density matching approach. *arXiv preprint arXiv:2507.12341*, 2025.

[55] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[56] Zhaiming Shen, Alexander Hsu, Rongjie Lai, and Wenjing Liao. Understanding in-context learning on structured manifolds: Bridging attention to kernel methods. In *International Conference on Learning Representations*, volume 2026, pages 42067–42103, 2026.

[57] Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. In *Forty-first International Conference on Machine Learning*, 2024.

[58] Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

[59] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, New York, NY, 2008.

[60] 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.

[61] Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.

[62] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[63] Rachel S.Y. Teo, Laziz Abdullaev, and Tan Minh Nguyen. The blessing and curse of dimensionality in safety alignment. In *Second Conference on Language Modeling*, 2025.

[64] Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen, editors, *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[65] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4344–4353, Hong Kong, China, November 2019. Association for Computational Linguistics.

[66] 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.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[68] Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

[69] Hieu M. Vu and Tan Minh Nguyen. Angular steering: Behavior control via rotation in activation space. In *2nd Workshop on Models of Human Feedback for AI Alignment*, 2025.

[70] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

[71] Jiaqing Xie. A comparative analysis of sparse autoencoder and activation difference in language model steering. In *arXiv preprint*, 2025.

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

[73] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*, 2023.

Supplement to "Kernelized Activation Steering"

Table of Contents

A Technical Derivations and Proofs **16**

A.1 Proofs . . . 16

A.2 Steering Gradients for Different Kernels . . . 18

A.3 Matrix Form of RBF-Kernel Steering . . . 20

A.4 Nyström Approximation . . . 20

B Additional Experimental Details **21**

B.1 Common Implementation Details . . . 21

B.2 Jailbreak Steering Details . . . 21

B.3 CAA-style Multiple-choice Steering Details . . . 21

B.4 Image Style-control Details . . . 22

C Additional Results **23**

C.1 Additional CAA-style Multiple-choice Results . . . 23

C.2 Additional Ablation Results . . . 23

C.3 Results on Cost Analysis . . . 24

C.4 Additional Image Style-control Qualitative Results . . . 24

D Broader Impacts **27**

E Use of Existing Assets **27**

A Technical Derivations and Proofs

A.1 Proofs

Proof of Theorem 4.4. Write Π in the form implied by (5),

$$\Pi(\boldsymbol{x}) \;=\; \frac{1}{n_B}\sum_{j=1}^{n_B} k(\boldsymbol{x}, \boldsymbol{b}_j) \;-\; \frac{1}{n_A}\sum_{i=1}^{n_A} k(\boldsymbol{x}, \boldsymbol{a}_i).$$

First, we upper bound the norm of $\nabla^2\Pi(\boldsymbol{x})$. Differentiation commutes with finite sums, so

$$\nabla^2\Pi(\boldsymbol{x}) \;=\; \frac{1}{n_B}\sum_{j} \nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{b}_j) \;-\; \frac{1}{n_A}\sum_{i} \nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{a}_i).$$

The operator norm is a norm on the space of symmetric matrices, so subadditivity (triangle inequality) and absolute homogeneity give

$$\|\nabla^2\Pi(\boldsymbol{x})\|_{\mathrm{op}} \;\le\; \frac{1}{n_B}\sum_{j}\|\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{b}_j)\|_{\mathrm{op}} + \frac{1}{n_A}\sum_{i}\|\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{a}_i)\|_{\mathrm{op}}.$$

By hypothesis each summand is at most H, so

$$\|\nabla^2\Pi(\boldsymbol{x})\|_{\mathrm{op}} \;\le\; H + H \;=\; 2H, \qquad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

Now note that since $\nabla^2 F(\boldsymbol{x}) = \nabla^2\Pi(\boldsymbol{x}) - \lambda I$,

$$\|\nabla^2 F(\boldsymbol{x})\|_{\mathrm{op}} \;\le\; \|\nabla^2\Pi(\boldsymbol{x})\|_{\mathrm{op}} + \lambda \;\le\; 2H + \lambda \;=:\; L.$$

Hence ∇F is L-Lipschitz as for any $\boldsymbol{x}, \boldsymbol{y}$, the fundamental theorem of calculus (FTC) gives

$$\nabla F(\boldsymbol{y}) - \nabla F(\boldsymbol{x}) \;=\; \int_0^1 \nabla^2 F\big(\boldsymbol{x} + t(\boldsymbol{y}-\boldsymbol{x})\big)(\boldsymbol{y}-\boldsymbol{x})\, dt,$$

and taking norms,

$$\|\nabla F(\boldsymbol{y}) - \nabla F(\boldsymbol{x})\| \;\le\; \int_0^1 \|\nabla^2 F\big(\boldsymbol{x} + t(\boldsymbol{y}-\boldsymbol{x})\big)\|_{\mathrm{op}}\, dt \cdot \|\boldsymbol{y}-\boldsymbol{x}\| \;\le\; L\|\boldsymbol{y}-\boldsymbol{x}\|. \tag{10}$$

Using the FTC again on F along the segment from $\boldsymbol{x}$ to $\boldsymbol{y}$,

$$F(\boldsymbol{y}) - F(\boldsymbol{x}) - \langle \nabla F(\boldsymbol{x}),\, \boldsymbol{y}-\boldsymbol{x}\rangle \;=\; \int_0^1 \langle \nabla F\big(\boldsymbol{x} + t(\boldsymbol{y}-\boldsymbol{x})\big) - \nabla F(\boldsymbol{x}),\, \boldsymbol{y}-\boldsymbol{x}\rangle\, dt.$$

Now Cauchy–Schwarz and inequality (10) give

$$\Big|F(\boldsymbol{y}) - F(\boldsymbol{x}) - \langle \nabla F(\boldsymbol{x}),\, \boldsymbol{y}-\boldsymbol{x}\rangle\Big| \;\le\; \int_0^1 Lt\|\boldsymbol{y}-\boldsymbol{x}\|^2\, dt \;=\; \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2,$$

so in particular

$$F(\boldsymbol{y}) \;\ge\; F(\boldsymbol{x}) + \langle \nabla F(\boldsymbol{x}),\, \boldsymbol{y}-\boldsymbol{x}\rangle - \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2. \tag{11}$$

Finally, let $\boldsymbol{g}_t := \nabla F(\boldsymbol{x}_t) \neq 0$ and apply (11) with $\boldsymbol{x} = \boldsymbol{x}_t$ and $\boldsymbol{y} = \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\, \boldsymbol{g}_t/\|\boldsymbol{g}_t\|$. Since $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| = \eta$ and $\langle \boldsymbol{g}_t,\, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle = \eta\|\boldsymbol{g}_t\|$, we obtain

$$F(\boldsymbol{x}_{t+1}) \ge F(\boldsymbol{x}_t) + \eta\|\boldsymbol{g}_t\| - \frac{L\eta^2}{2}.$$

Therefore, strict ascent is guaranteed whenever $0 < \eta < 2\|\boldsymbol{g}_t\|/L$. Taking $L = 2H + \lambda$ gives the sufficient condition $0 < \eta < \|\nabla F(\boldsymbol{x}_t)\|/(H + \lambda/2)$. □

Proof of Corollary 4.5. Recall $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\!\big(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/(2\sigma^2)\big)$.

Now the proof of Corollary follows from direct computation. Let $\boldsymbol{v} := \boldsymbol{y} - \boldsymbol{x}$. From $\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) = \sigma^{-2}\boldsymbol{v}\, k(\boldsymbol{x}, \boldsymbol{y})$ and the product rule,

$$\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) = \nabla_{\boldsymbol{x}}\Big[\sigma^{-2}\boldsymbol{v}\, k(\boldsymbol{x}, \boldsymbol{y})\Big] = \sigma^{-2}\Big[-I \cdot k(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{v}\, \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y})^\top\Big] = \frac{k(\boldsymbol{x}, \boldsymbol{y})}{\sigma^2}\left[\frac{\boldsymbol{v}\boldsymbol{v}^\top}{\sigma^2} - I\right].$$

Note that the bracketed matrix $M := \sigma^{-2}\boldsymbol{v}\boldsymbol{v}^\top - I$ is symmetric. Its spectrum is $\{\sigma^{-2}\|\boldsymbol{v}\|^2 - 1\}$ on $\mathrm{span}(\boldsymbol{v})$ together with $\{-1\}$ on $\boldsymbol{v}^\perp$ (dimension $d-1$). Hence

$$\|M\|_{\mathrm{op}} = \max\Big(\big|\sigma^{-2}\|\boldsymbol{v}\|^2 - 1\big|,\, 1\Big) \;\le\; \frac{\|\boldsymbol{v}\|^2}{\sigma^2} + 1,$$

where the inequality uses $|a-1| \le a+1$ for $a \ge 0$ and trivially $1 \le a+1$. Combining with $0 \le k(\boldsymbol{x}, \boldsymbol{y}) \le 1$,

$$\|\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y})\|_{\mathrm{op}} \;\le\; \frac{1}{\sigma^2}\left(\frac{\|\boldsymbol{y}-\boldsymbol{x}\|^2}{\sigma^2} + 1\right). \tag{12}$$

If $\boldsymbol{x}, \boldsymbol{y}$ both lie in a ball of radius R, then $\|\boldsymbol{y}-\boldsymbol{x}\| \le 2R$, so $\|\boldsymbol{y}-\boldsymbol{x}\|^2 \le 4R^2$. Substituting into (12),

$$\|\nabla^2_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y})\|_{\mathrm{op}} \;\le\; \frac{4R^2}{\sigma^4} + \frac{1}{\sigma^2} \;=:\; H.$$

Substituting this value of H into the normalized-step condition of Theorem 4.4 gives

$$0 < \eta < \frac{\|\nabla F(\boldsymbol{x}_t)\|}{\frac{4R^2}{\sigma^4} + \frac{1}{\sigma^2} + \frac{\lambda}{2}}.$$

□

A.2 Steering Gradients for Different Kernels

The particular form of the KAS update is determined by the gradient of the chosen kernel with respect to its first argument. Recall that the empirical steering potential is defined as

$$\Pi(\boldsymbol{x}) = \frac{1}{n_B}\sum_{j=1}^{n_B} k(\boldsymbol{x}, \boldsymbol{b}_j) - \frac{1}{n_A}\sum_{i=1}^{n_A} k(\boldsymbol{x}, \boldsymbol{a}_i). \tag{13}$$

Whenever the kernel is differentiable with respect to $\boldsymbol{x}$, differentiation may be applied term by term, yielding

$$\nabla_{\boldsymbol{x}}\Pi(\boldsymbol{x}) = \frac{1}{n_B}\sum_{j=1}^{n_B} \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{b}_j) - \frac{1}{n_A}\sum_{i=1}^{n_A} \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{a}_i). \tag{14}$$

The resulting steering update is therefore

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta\left[\nabla_{\boldsymbol{x}}\Pi(\boldsymbol{x}) - \lambda(\boldsymbol{x} - \boldsymbol{x}_0)\right]. \tag{15}$$

We now derive this gradient explicitly for several commonly used kernels.

Gaussian RBF kernel. Consider first the Gaussian radial basis function kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}\right), \tag{16}$$

where $\sigma > 0$ denotes the bandwidth. Differentiating with respect to $\boldsymbol{x}$ gives

$$\begin{aligned}\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) &= k(\boldsymbol{x}, \boldsymbol{y})\nabla_{\boldsymbol{x}}\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}\right) \\ &= -\frac{1}{\sigma^2} k(\boldsymbol{x}, \boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y}) \\ &= \frac{1}{\sigma^2} k(\boldsymbol{x}, \boldsymbol{y})(\boldsymbol{y} - \boldsymbol{x}).\end{aligned} \tag{17}$$

Substitution into (14) yields

$$\nabla_{\boldsymbol{x}}\Pi_{\mathrm{RBF}}(\boldsymbol{x}) = \frac{1}{\sigma^2}\left[\frac{1}{n_B}\sum_{j=1}^{n_B} k(\boldsymbol{x}, \boldsymbol{b}_j)(\boldsymbol{b}_j - \boldsymbol{x}) - \frac{1}{n_A}\sum_{i=1}^{n_A} k(\boldsymbol{x}, \boldsymbol{a}_i)(\boldsymbol{a}_i - \boldsymbol{x})\right]. \tag{18}$$

Thus each target reference activation contributes an attractive force pointing from $\boldsymbol{x}$ toward $\boldsymbol{b}_j$, whereas each source activation contributes a repulsive force. The magnitude of each contribution is modulated by its kernel similarity to the current activation, making the resulting steering field local and activation dependent.

Degree-two polynomial kernel. We next consider the degree-two polynomial kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y} + c)^2, \tag{19}$$

where $c \geq 0$ is an offset parameter. By the chain rule,

$$\begin{aligned}\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) &= 2(\boldsymbol{x}^\top \boldsymbol{y} + c)\nabla_{\boldsymbol{x}}(\boldsymbol{x}^\top \boldsymbol{y}) \\ &= 2(\boldsymbol{x}^\top \boldsymbol{y} + c)\boldsymbol{y}.\end{aligned} \tag{20}$$

Consequently, the corresponding KAS steering field is

$$\nabla_{\boldsymbol{x}}\Pi_{\mathrm{poly}}(\boldsymbol{x}) = \frac{2}{n_B}\sum_{j=1}^{n_B}(\boldsymbol{x}^\top \boldsymbol{b}_j + c)\boldsymbol{b}_j - \frac{2}{n_A}\sum_{i=1}^{n_A}(\boldsymbol{x}^\top \boldsymbol{a}_i + c)\boldsymbol{a}_i. \tag{21}$$

The structure of this gradient becomes more transparent when written in terms of empirical moments. Define the empirical means

$$\boldsymbol{\mu}_B = \frac{1}{n_B}\sum_{j=1}^{n_B} \boldsymbol{b}_j, \qquad \boldsymbol{\mu}_A = \frac{1}{n_A}\sum_{i=1}^{n_A} \boldsymbol{a}_i,$$

and the uncentered second-moment matrices

$$M_B = \frac{1}{n_B}\sum_{j=1}^{n_B} \boldsymbol{b}_j \boldsymbol{b}_j^\top, \qquad M_A = \frac{1}{n_A}\sum_{i=1}^{n_A} \boldsymbol{a}_i \boldsymbol{a}_i^\top.$$

Using the identity $(\boldsymbol{x}^\top \boldsymbol{y})\boldsymbol{y} = \boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{x}$, the target contribution can be rewritten as

$$\frac{2}{n_B}\sum_{j=1}^{n_B} (\boldsymbol{x}^\top \boldsymbol{b}_j + c)\boldsymbol{b}_j = 2M_B \boldsymbol{x} + 2c\boldsymbol{\mu}_B, \tag{22}$$

and similarly for the source contribution. Hence,

$$\nabla_{\boldsymbol{x}} \Pi_{\mathrm{poly}}(\boldsymbol{x}) = 2(M_B - M_A)\boldsymbol{x} + 2c(\boldsymbol{\mu}_B - \boldsymbol{\mu}_A). \tag{23}$$

Equation (23) gives a useful interpretation of degree-two polynomial steering. The first term depends on the difference between the second moments of the target and source activation distributions, while the second term is proportional to the ordinary Difference-in-Means direction. Thus the polynomial kernel extends linear steering by incorporating second-order geometric information.

In particular, when $c = 0$,

$$\nabla_{\boldsymbol{x}} \Pi_{\mathrm{poly}}(\boldsymbol{x}) = 2(M_B - M_A)\boldsymbol{x}, \tag{24}$$

so steering remains possible even when the two reference sets have identical means but different second-order structure. This contrasts with linear DiM, whose population signal vanishes whenever the two distributions have the same mean. When $c > 0$, the polynomial field combines this second-order correction with a global mean-difference component.

Laplacian kernel. Finally, consider the Laplacian kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|_1}{\sigma}\right), \tag{25}$$

where $\sigma > 0$. Since

$$\|\boldsymbol{x} - \boldsymbol{y}\|_1 = \sum_{r=1}^{d} |x_r - y_r|, \tag{26}$$

its derivative with respect to coordinate x_r, whenever $x_r \neq y_r$, is

$$\frac{\partial}{\partial x_r}\|\boldsymbol{x} - \boldsymbol{y}\|_1 = \mathrm{sign}(x_r - y_r). \tag{27}$$

Applying the chain rule therefore gives

$$\begin{aligned} \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) &= -\frac{1}{\sigma} k(\boldsymbol{x}, \boldsymbol{y})\,\mathrm{sign}(\boldsymbol{x} - \boldsymbol{y}) \\ &= \frac{1}{\sigma} k(\boldsymbol{x}, \boldsymbol{y})\,\mathrm{sign}(\boldsymbol{y} - \boldsymbol{x}), \end{aligned} \tag{28}$$

where the sign function is applied elementwise.

Substitution into the general KAS gradient gives

$$\nabla_{\boldsymbol{x}} \Pi_{\mathrm{Lap}}(\boldsymbol{x}) = \frac{1}{\sigma}\left[\frac{1}{n_B}\sum_{j=1}^{n_B} k(\boldsymbol{x}, \boldsymbol{b}_j)\,\mathrm{sign}(\boldsymbol{b}_j - \boldsymbol{x}) - \frac{1}{n_A}\sum_{i=1}^{n_A} k(\boldsymbol{x}, \boldsymbol{a}_i)\,\mathrm{sign}(\boldsymbol{a}_i - \boldsymbol{x})\right]. \tag{29}$$

Unlike the RBF kernel, the Laplacian kernel does not scale each coordinate of the steering direction by the magnitude of the displacement $|y_r - x_r|$. Instead, away from nondifferentiable coordinates, each coordinate contributes only through its sign, while the overall strength of a reference point is controlled by the exponentially decaying kernel weight. Consequently, the Laplacian field produces a coordinate-wise directional pull toward target samples and away from source samples.

The Laplacian kernel is not differentiable at coordinates satisfying $x_r = y_r$. At such points, the ℓ_1 norm admits a subgradient, and we use the conventional choice

$$\mathrm{sign}(0) = 0. \tag{30}$$

Accordingly, (28) and (29) should be interpreted as subgradient expressions at nondifferentiable points.

A.3 Matrix Form of RBF-Kernel Steering

Deriving the matrix form for our formulas is useful for GPU-friendly implementation. Let $X = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_N^\top]^\top \in \mathbb{R}^{N\times d}$ denote the token activations in a transformer layer, and $A \in \mathbb{R}^{n_A\times d}$, $B \in \mathbb{R}^{n_B\times d}$ be the reference activations for source and target datasets, respectively. For the RBF kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right),$$

the pairwise kernel matrices are

$$K_B = \exp\left(-\frac{\|X\|^2_{\text{row}}\mathbf{1}_{n_B}^\top + \mathbf{1}_N\|B\|^{2\top}_{\text{row}} - 2XB^\top}{2\sigma^2}\right),$$

$$K_A = \exp\left(-\frac{\|X\|^2_{\text{row}}\mathbf{1}_{n_A}^\top + \mathbf{1}_N\|A\|^{2\top}_{\text{row}} - 2XA^\top}{2\sigma^2}\right),$$

where $\|X\|^2_{\text{row}} \in \mathbb{R}^N$ contains row-wise squared norms of X.

Define the kernel-weighted means and row-sum vectors:

$$W_B = K_B B, \quad W_A = K_A A, \quad s_B = K_B\mathbf{1}_{n_B}, \quad s_A = K_A\mathbf{1}_{n_A}.$$

Then the matrix-form gradient of the kernel steering potential $\Pi(X)$ is (see the RBF gradient in the previous section)

$$\nabla_X\Pi(X) = \frac{1}{\sigma^2}\left[\frac{1}{n_B}W_B - \frac{1}{n_A}W_A - \text{Diag}\left(\frac{s_B}{n_B} - \frac{s_A}{n_A}\right)X\right].$$

The joint update for all N tokens with step size η and regularization λ is

$$X \leftarrow X + \eta\left[\nabla_X\Pi(X) - \lambda(X - X_0)\right].$$

A.4 Nyström Approximation

We derive the kernel approximation considered in Section 5.4. Let $Z = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m\} \subseteq A \cup B$ be a fixed set of landmarks, with $m \ll n_A + n_B$. Define the landmark Gram matrix $W \in \mathbb{R}^{m\times m}$ and kernel vector $\boldsymbol{k}_Z(\boldsymbol{x}) \in \mathbb{R}^m$ by

$$W_{\ell r} = k(\boldsymbol{z}_\ell, \boldsymbol{z}_r), \qquad \boldsymbol{k}_Z(\boldsymbol{x}) = \big[k(\boldsymbol{x}, \boldsymbol{z}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{z}_m)\big]^\top. \tag{31}$$

The Nyström approximation [70] is

$$\widetilde{k}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{k}_Z(\boldsymbol{x})^\top W^\dagger \boldsymbol{k}_Z(\boldsymbol{y}), \tag{32}$$

where $W^\dagger$ is the Moore–Penrose pseudoinverse. Since W is positive semidefinite, (32) defines a positive semidefinite kernel with feature map $\widetilde{\varphi}(\boldsymbol{x}) = (W^\dagger)^{1/2}\boldsymbol{k}_Z(\boldsymbol{x})$.

Approximate steering score. Replacing k by $\widetilde{k}$ in (5) gives

$$\begin{aligned}\widetilde{\Pi}(\boldsymbol{x}) &= \frac{1}{n_B}\sum_{j=1}^{n_B}\widetilde{k}(\boldsymbol{x}, \boldsymbol{b}_j) - \frac{1}{n_A}\sum_{i=1}^{n_A}\widetilde{k}(\boldsymbol{x}, \boldsymbol{a}_i) \\ &= \boldsymbol{k}_Z(\boldsymbol{x})^\top \boldsymbol{c}, \qquad \boldsymbol{c} = W^\dagger \boldsymbol{d}_Z,\end{aligned} \tag{33}$$

where

$$\boldsymbol{d}_Z = \frac{1}{n_B}\sum_{j=1}^{n_B}\boldsymbol{k}_Z(\boldsymbol{b}_j) - \frac{1}{n_A}\sum_{i=1}^{n_A}\boldsymbol{k}_Z(\boldsymbol{a}_i). \tag{34}$$

The coefficients $\boldsymbol{c}$ are precomputed for each fixed layer, reference set, and kernel configuration. All reference activations contribute to (34); the approximation does not replace the reference sets by the landmarks alone.

Table 5: Common implementation details used across experiments.

Item	Setting
Intervention space	Residual stream activations
Default token position	Last prompt token
Kernel type	Gaussian RBF
RBF bandwidth	Median heuristic [21]
Regularizer λ	0

Gradient and normalized update. Holding the landmarks and coefficients fixed yields

$$\nabla\widetilde{\Pi}(\boldsymbol{x}) = \sum_{\ell=1}^{m} c_\ell \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{z}_\ell). \tag{35}$$

For the RBF kernel (8), this becomes

$$\nabla\widetilde{\Pi}(\boldsymbol{x}) = \frac{1}{\sigma^2} \sum_{\ell=1}^{m} c_\ell k(\boldsymbol{x}, \boldsymbol{z}_\ell)(\boldsymbol{z}_\ell - \boldsymbol{x}). \tag{36}$$

Equivalently, let $Z \in \mathbb{R}^{m\times d}$ contain the landmark activations as rows and set $\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{c} \odot \boldsymbol{k}_Z(\boldsymbol{x})$, where $\odot$ denotes elementwise multiplication. Then

$$\nabla\widetilde{\Pi}(\boldsymbol{x}) = \frac{1}{\sigma^2} \left[Z^\top \boldsymbol{w}(\boldsymbol{x}) - \left(\mathbf{1}^\top \boldsymbol{w}(\boldsymbol{x})\right)\boldsymbol{x} \right]. \tag{37}$$

We retain the proximal objective and normalized update of Section 4.2:

$$\begin{aligned} \widetilde{F}(\boldsymbol{x}) &= \widetilde{\Pi}(\boldsymbol{x}) - \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2, \qquad \widetilde{\boldsymbol{g}}_t = \nabla\widetilde{\Pi}(\boldsymbol{x}_t) - \lambda(\boldsymbol{x}_t - \boldsymbol{x}_0), \\ \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t + \eta \frac{\widetilde{\boldsymbol{g}}_t}{\|\widetilde{\boldsymbol{g}}_t\|}, \qquad \widetilde{\boldsymbol{g}}_t \neq 0. \end{aligned} \tag{38}$$

B Additional Experimental Details

This appendix provides additional implementation details, hyperparameters, dataset construction procedures, and supplementary results for the experiments in Section 5.

B.1 Common Implementation Details

Unless otherwise stated, all activation interventions are applied to the residual stream at the final prompt-token position. For KAS, we use the RBF kernel with bandwidth selected by the median heuristic [21] over the source and target reference activations. We set the regularizer to $\lambda = 0$. Unless otherwise stated, KAS uses a single gradient-ascent step at inference time. Multi-step KAS results are reported explicitly when used, with the number of steps specified in the corresponding table, figure, or caption.

B.2 Jailbreak Steering Details

For jailbreak steering, harmful prompts are sampled from ADVBENCH, MALICIOUSINSTRUCT, TDC2023, and HARMBENCH, while harmless prompts are sampled from ALPACA. We use 128 harmful and 128 harmless prompts to construct the contrastive reference sets in the main experiments. Evaluation is performed on JAILBREAKBENCH using LLAMA-GUARD-3-8B to compute attack success rate (ASR). General utility is evaluated using TINYBENCHMARKS. For each model, ActAdd, Curveball, and KAS are applied at the same intervention layer.

B.3 CAA-style Multiple-choice Steering Details

For CAA-style evaluation, each contrastive pair shares the same multiple-choice question but differs only in the appended answer option. Positive examples append the behavior-consistent answer,

Table 6: Curveball hyperparameters used in the jailbreak experiments.

Item	Setting
Kernel	Polynomial kernel, $k(x, y) = (x^\top y + \gamma)^p$
Polynomial degree p	$p = 3$
Kernel bias γ	$\gamma = 1$
Bias term `coef0`	Grid over 1.0
KPCA dimension m	$m = 20$

Table 7: Curveball layer and steering-strength selection for jailbreak experiments.

Model	Layer	α
GEMMA-2-9B-IT	23	20.0
LLAMA-3.1-8B-INSTRUCT	13	150.0
QWEN2.5-7B-INSTRUCT	18	10.0

Table 8: KAS hyperparameters for jailbreak steering experiments.

Model	Layer	KAS step size η
GEMMA-2-9B-IT	22	200.0
LLAMA-3.1-8B-INSTRUCT	12	5.0
QWEN2.5-7B-INSTRUCT	18	10.0

whereas negative examples append the opposite answer. Activations are extracted at the answer-token position. We evaluate on 50 held-out questions per behavior and report the average probability assigned to the behavior-consistent option.

Dataset sourcing. The behavioral datasets span several alignment-relevant attributes: AI Coordination, Corrigibility, Hallucination, Myopic Reward, Survival Instinct, and Sycophancy. Except for Hallucination, these datasets are drawn primarily from Anthropic's Advanced AI Risk human-written evaluations [47]. The Hallucination dataset is constructed separately from GPT-4-generated contrastive multiple-choice questions.

System-prompt conditions. We evaluate three prompting conditions: no system prompt, a positive system prompt that encourages the target behavior, and a negative system prompt that discourages the target behavior. These settings allow us to test whether steering remains effective when the prompt context is neutral, aligned with the target concept, or opposed to it.

Hyperparameter selection. For the CAA-style multiple-choice experiments, all methods are evaluated on the models GEMMA-2-9B-IT, LLAMA-3.1-8B-INSTRUCT and QWEN2.5-7B-INSTRUCT. We sweep the intervention layer over 22-23 for GEMMA-2-9B-IT, 12-13 for LLAMA-3.1-8B-INSTRUCT and 17-18 for QWEN2.5-7B-INSTRUCT. In addition, we select the best configuration according to the average token probability assigned to the behavior-consistent answer on the validation set. We then evaluate the selected configuration on the disjoint test set. For both ActAdd and KAS, we sweep the steering strength η over 0.01, 0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.25, 3.5, 3.75, 4.0, 4.25, 4.5, 4.75, 5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0, 22.5, 25.0, 27.5, 30.0, 32.5, 35.0, 37.5, 40.0, 42.5, 45.0, 47.5, 50.0, 55.0, 60.0, 62.5, 65.0, 70.0, 75.0, 80.0, 85.0, 87.5, 90.0, 95.0, 100.0, 112.5, 125.0, 137.5, 150.0, 162.5, 175.0, 187.5, 200.0, 225.0, 250.0, 275.0, 300.0, 325.0, 350.0, 355.0, 360.0, 365.0, 370.0, 375.0, 380.0, 385.0, 390.0, 395.0, 400.0, 405.0, 410.0, 420.0, 425.0, 430.0, 435.0, 440.0, 445.0, 450.0, 500.0, 625.0, 750.0. We present the chosen hyperparameter set for Curveball and KAS in Table 6, Table 7 and Table 8.

B.4 Image Style-control Details

For image style control, we sample prompts from COCO Captions and construct source–target activation pairs using original and style-augmented prompts. Images are generated using FLUX.1-SCHNELL. We evaluate style induction using zero-shot CLIP classification and content preservation using CLIPScore against the original prompt.

Table 9: CAA-style steering results under the negative system-prompt condition. We report the average token probability assigned to the behavior-consistent answer. "No steering" denotes the unsteered baseline, while ActAdd and KAS report the best steered result together with the selected step size η and intervention layer. The best probability in each row is shown in bold.

Behavior	No steering	ActAdd			KAS		
	Prob.	Prob.	η	Layer	Prob.	η	Layer
AI Coordination	8.69	39.51	2.50	23	**40.19**	300.00	23
Corrigibility	0.05	33.82	2.50	23	**43.45**	275.00	22
Hallucination	23.97	43.97	2.50	23	**55.01**	400.00	22
Myopic Reward	1.29	49.56	5.00	22	**58.48**	435.00	23
Survival Instinct	43.68	43.56	0.01	23	**43.83**	0.01	23
Sycophancy	53.18	53.47	0.25	22	**53.69**	17.50	22

C Additional Results

C.1 Additional CAA-style Multiple-choice Results

The main paper reports CAA-style steering results under the negative system-prompt condition, where the system instruction discourages the target behavior. For completeness, we provide the full selected hyperparameters for this condition, together with additional results under the no-system-prompt and positive system-prompt conditions in Tables 9–11. All experiments follow the setup in Section 5.2: steering is performed on GEMMA-2-9B-IT, activations are extracted at the answer-token position, and performance is measured by the average token probability assigned to the behavior-consistent answer.

Table 10: CAA-style steering results under the no-system-prompt condition. We report the average token probability assigned to the behavior-consistent answer. "No steering" denotes the unsteered baseline, while ActAdd and KAS report the best steered result together with the selected step size η and intervention layer. The best probability in each row is shown in bold.

Behavior	No steering	ActAdd			KAS		
	Prob.	Prob.	η	Layer	Prob.	η	Layer
AI Coordination	6.31	33.23	2.50	23	**35.61**	400.00	22
Corrigibility	43.50	46.84	2.50	23	**47.42**	250.00	23
Hallucination	28.79	47.23	2.50	23	**54.79**	400.00	22
Myopic Reward	23.62	60.11	4.50	23	**60.32**	400.00	23
Survival Instinct	67.09	66.98	0.01	23	**67.11**	0.10	22
Sycophancy	**53.44**	53.26	0.01	22	53.40	0.01	23

Across all system-prompt conditions, KAS attains the best probability for nearly all behaviors and matches the strongest baseline in the remaining saturated cases. These results indicate that KAS provides robust concept control across neutral, supportive, and opposing prompt contexts.

C.2 Additional Ablation Results

In this section, we provide additional ablation results on specified hyperparameters of KAS on the jailbreaking task. Unless otherwise stated, all experiments in this section use the setups specified in Section B.2.

Kernel choice. The main paper described how KAS supports different kernels, as well as how different kernels can benefit KAS in different ways. We provide the ablation results on kernel choice in Table 12, in which experiments are conducted on 3 different kernels - namely Laplacian, polynomial and RBF. The polynomial kernel is shown to performs comparably to the RBF kernel, while Laplacian kernel inducted a fall in attack success rate. This result suggests that KAS performance is indeed dependent on the kernel choice.

Number of data instances. In Section 5.5 and Figure 5 we demonstrated the ability of KAS to perform qualitative steering using a single pair of contrastive data points, in comparison with other steering methods. In Table 13 we further show how KAS improves as it is given more data. The

Table 11: CAA-style steering results under the positive system-prompt condition. We report the average token probability assigned to the behavior-consistent answer. "No steering" denotes the unsteered baseline, while ActAdd and KAS report the best steered result together with the selected step size η and intervention layer. The best probability in each row is shown in bold.

Behavior	**No steering**	**ActAdd**			**KAS**		
	Prob.	**Prob.**	η	**Layer**	**Prob.**	η	**Layer**
AI Coordination	8.42	37.13	2.50	23	**38.50**	300.00	23
Corrigibility	72.32	72.11	0.01	22	**72.46**	0.01	22
Hallucination	19.59	44.53	2.50	23	**53.49**	400.00	22
Myopic Reward	99.99	99.99	0.01	22	**99.99**	0.10	23
Survival Instinct	90.67	90.44	0.01	22	**90.71**	0.10	22
Sycophancy	56.50	56.50	0.01	22	**56.59**	0.01	23

Table 12: KAS attack success rate and perplexity across kernels on jailbreaking task.

Kernel	Prompt only	KAS ASR	KAS Perplexity
Laplacian	False	0.70	7.484
Laplacian	True	0.59	7.484
Polynomial	False	0.93	8.149
Polynomial	True	0.81	8.149
RBF	False	0.96	8.343
RBF	True	0.88	8.343

number of contrastive data pairs are chosen from the set $\{1, 2, 4, 8, 16, 32, 64, 128\}$. We observed that while KAS can achieve notable ASR on datasets as small as 2 contrastive pairs, its performance can still be improved with more data provided.

Table 13: KAS attack success rate and perplexity as the number of reference examples grows.

Num. of references (n_{train})	KAS ASR	KAS Perplexity
1	0.43	6.258
2	0.88	7.141
4	0.93	5.716
8	0.88	6.104
16	0.96	6.488
32	0.96	6.542
64	0.95	6.363
128	0.97	6.328

C.3 Results on Cost Analysis

In the main paper, we discussed the KAS procedure and its comparison against no steering setups as well as other steering methods (mainly ActAdd). In this part, we performed an analysis on runtime of KAS compared to those setups. Experiments are performed on a single NVIDIA H100 GPU on 50 samples from the CAA dataset of myopic reward, recorded with one step of steering. Table 14 report our results.

From these results, we can see that KAS incurs minimal overhead in this measurement of execution time compared to the ActAdd and no steering setups, highlighting the effectiveness of KAS in resource consumption.

C.4 Additional Image Style-control Qualitative Results

The main paper reports style-control results for the *cyberpunk* style, measuring style induction with 0-shot classification and content preservation with CLIPSCORE. Here, we provide additional qualitative results across more prompts for *cyberpunk* and *impressionism* styles in Figures 6 and 7.

Table 14: Inference time overhead of steering methods.

Methodology	Time elapsed (s)	Overhead vs. no steering
No steering	11.44	0%
ActAdd	11.48	0.35%
KAS	11.61	1.49%

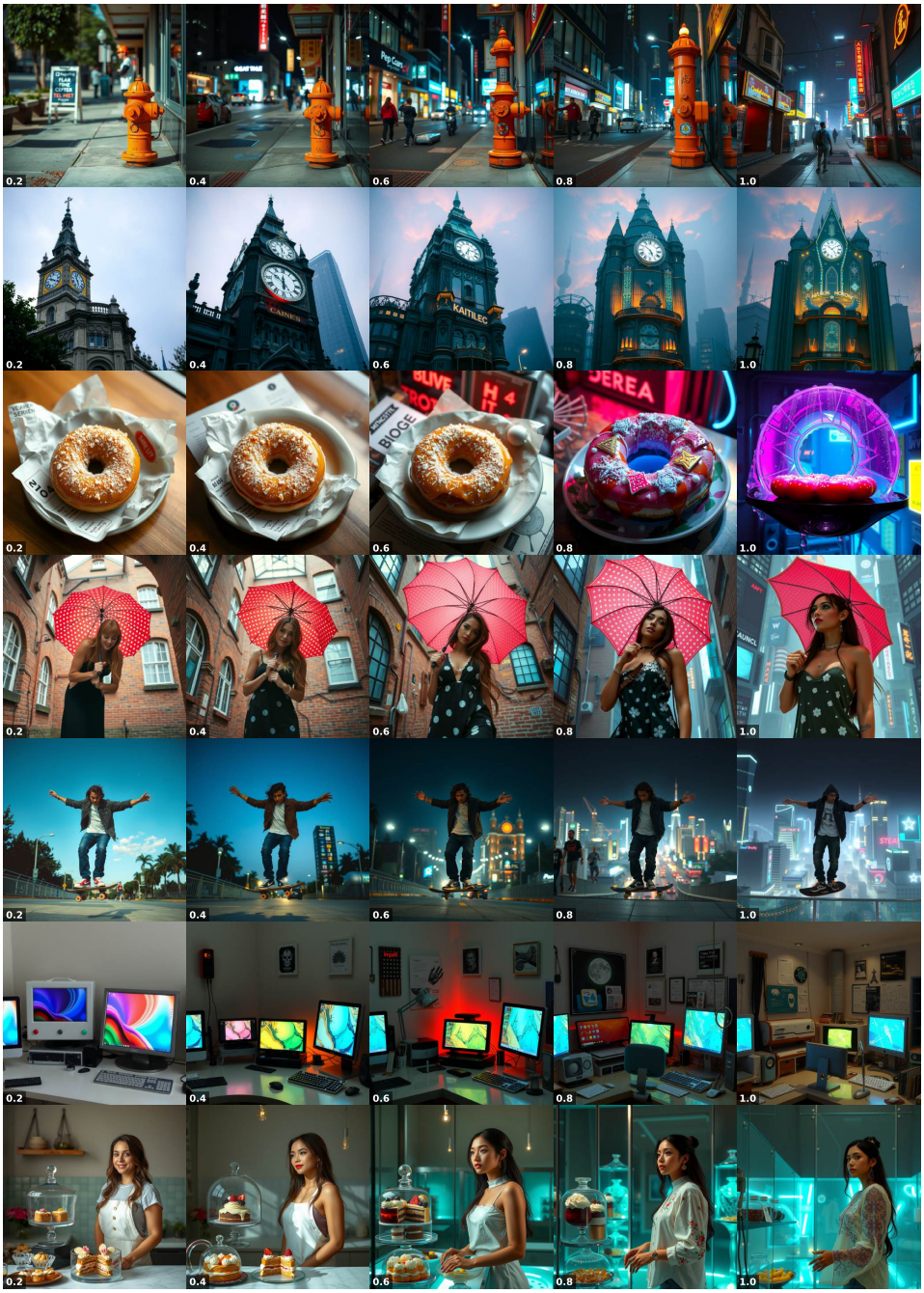


Figure 6: Qualitative examples of KAS style control for the *cyberpunk* style under increasing intervention strengths $\eta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.



Figure 7: Qualitative examples of KAS style control for the *impressionism* style under increasing intervention strengths $\eta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

Table 15: Existing model and evaluator assets used in this paper.

Asset	Variants used	Usage	License / terms
GEMMA 2 [20]	GEMMA-2-9B-IT	Target model	Gemma Terms of Use
LLAMA 3.1	LLAMA-3.1-8B-INSTRUCT	Target model	Llama 3.1 Community License
QWEN2.5 [49]	QWEN2.5-7B-INSTRUCT	Target model	Apache-2.0
META LLAMA GUARD 3 [34]	LLAMA-GUARD-3-8B	Evaluator	Llama 3.1 Community License
FLUX [29]	FLUX.1 [SCHNELL]	Target model	Apache-2.0

Table 16: Existing dataset and benchmark assets used in this paper.

Asset	License / terms
ADVBENCH [73]	MIT
MALICIOUSINSTRUCT [26]	CC BY-SA 4.0
TDC2023 [37]	MIT
HARMBENCH [36]	MIT
ALPACA [60]	CC BY-NC 4.0
JAILBREAKBENCH [12]	MIT
THE PILE [19]	MIT
TRUTHFULQA [32]	Apache-2.0
TINYBENCHMARKS [48]	MIT
ANTHROPIC MODEL-WRITTEN EVALUATIONS [47]	CC-BY-4.0
COCO CAPTIONS [13]	CC BY 4.0

D Broader Impacts

Kernelized Activation Steering (KAS) provides a training-free mechanism for modifying model behavior at inference time. This has potential positive applications in controllability, safety evaluation, and model debugging. By enabling behavior-specific interventions without updating model parameters, KAS may help practitioners study how target attributes are represented in internal activations, evaluate the robustness of safety mechanisms, and develop more fine-grained tools for aligning model outputs with user- or deployment-specific requirements. More broadly, geometry-aware steering methods may contribute to more interpretable and adaptive forms of model control than global linear interventions.

At the same time, activation steering is inherently dual-use. The same mechanisms that can suppress undesirable behavior or improve controllability can also be used to elicit harmful behavior, bypass refusal mechanisms, or manipulate model responses in unintended ways. This concern is particularly relevant for our jailbreak experiments. These experiments are included to measure controllability and expose weaknesses in current alignment mechanisms, not to encourage misuse. We therefore view KAS as both a potential safety tool and a method that should be handled with care.

To reduce misuse risk, we emphasize the need for appropriate safeguards, clear documentation of intended use, and accountability in deployment. Practitioners applying KAS beyond controlled research settings should perform downstream safety evaluations before release. We further encourage continued ethical review of inference-time steering methods to ensure that their use remains aligned with safety, transparency, and responsible deployment practices.

E Use of Existing Assets

This section summarizes the existing models, datasets, and benchmark resources used in this work, along with their corresponding licenses or terms of use in Tables 15 and 16.

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 abstract and introduction state the main contributions of KAS: a training-free, kernelized generalization of activation steering, a closed-form RBF instantiation, and a theoretical connection to DiM. The empirical claims are limited to the evaluated jailbreak, CAA-style multiple-choice, and image style-control settings.

   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: Limitations of this work is discussed in Section 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 provide the full set of assumptions and complete (and correct) proofs for the main theorems in Appendix A.1.

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: The paper specifies the evaluated models, datasets, baselines, metrics, intervention layers, steering settings, and hyperparameter search procedures. Additional implementation details and hyperparameters are provided in Appendix B.

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 open access to the data and code in Supplementary Material, with sufficient instructions to faithfully reproduce the main experimental results.

   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: The paper describes the proposed method, datasets, models, steering settings, metrics, and hyperparameter sweeps used in the main experiments and ablaion studies. Additional experimental details are provided in the appendix.

   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: Section 5.5 reports summary statistics across prompt-pair samples, including mean and standard deviation where applicable.

   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 provide sufficient information on the computer resources needed to reproduce the experiments in Section 5.

   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: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   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 broader impacts in Appendix D.

    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: [N/A]

    Justification: The paper does not release any data, pre-trained models, image generators, or scraped datasets. Therefore, safeguards for responsible release are not applicable.

    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: The paper cites the creators of the existing datasets, benchmarks, models used in the experiments. The list of existing assets used is included in Appendix E.

    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 include details about training and implementation as well as limitations and code for our proposed method.

    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 experiments or 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 study participants or human-subject research. Therefore, participant risks, risk disclosure, and IRB approval are not applicable.

    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: LLMs are not used as an important, original, or non-standard component of the core methodology. Any use of LLMs is limited to writing, editing, or formatting assistance and does not affect the scientific content, 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.