ACTIVATION STEERING BY PATH LAW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

We develop a theoretical framework for activation steering based on matching activation path laws. Distinct path laws can share identical activation marginals at every layer, so marginal alignment alone cannot identify the intended path law. Using Kolmogorov's extension theorem, we characterize this law through consistent finite dimensional joint distributions. For finite depth networks, we construct the Kolmogorov loss to match joint activation distributions over contiguous groups of layers. Under the stated kernel conditions, vanishing population loss is equivalent to weak convergence toward the reference path law. With shared inputs and additional regularity assumptions, sufficiently accurate matching of actual intervened trajectories improves output agreement over exact marginal matching that retains positive functional error. We also establish conditions under which the difference-in-means vector is the unique optimal additive map. For practical fitting, we learn steering maps from cached activations without additional forward or backward passes through the pretrained model. Experiments on jailbreak and truthfulness across language models with 7B to 31B parameters demonstrate the framework's utility across several steering map families. [1]

1 INTRODUCTION

Large language models (LLMs) support conversational assistants (OpenAI, 2023), automated coding (Yang et al., 2024), and, with formal verification, systems that resolve selected open mathematical conjectures (Tsoukalas et al., 2026). Controlling their behavior while preserving these capabilities is therefore an important goal (Ouyang et al., 2022). Fine tuning can adapt behavior, but even parameter efficient methods such as LoRA require training through the model (Hu et al., 2021). Prompting avoids parameter updates, although longer prompts increase inference costs (Jiang et al., 2023b).

Activation steering instead modifies internal activations while keeping pretrained parameters fixed, using maps such as translations (Rimsky et al., 2024), projections (Arditi et al., 2024), or rotations (Vu & Nguyen, 2025). This provides a tractable framework for studying how changes to internal representations affect model behavior. Yet these interventions can also affect unrelated capabilities (Vu & Nguyen, 2025). What distributional target should guide steering maps, and when does matching it imply agreement with the desired behavior?

Existing distributional approaches use different targets. Activation Transport (Rodríguez et al., 2025a) estimates coordinatewise maps sequentially, accounting for earlier interventions without explicitly matching joint activation distributions across layers. MiMiC matches means and covariances (Singh et al., 2024), leaving higher order differences undetermined. LinEAS jointly learns parameters by backpropagating through the frozen backbone, accounting for downstream effects but comparing layerwise coordinate marginals and requiring repeated backbone passes (Rodríguez et al., 2025b). MAT-STEER uses maximum mean discrepancy (MMD) for multi attribute control at a selected layer, managing attribute interference without explicitly matching dependencies across depths (Nguyen et al., 2025).

In this work, we develop a theoretical framework for activation steering based on matching activation path laws defined as distributions over activation trajectories. Two networks can share every layer's activation distribution yet produce different outputs for the same input, because marginals discard dependencies across depths. This ambiguity persists when interventions are optimized jointly using

[1] Code: anonymous repository.

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

only layerwise objectives. Our framework separates the reference law, steering map family, and discrepancy, derives a matching objective, and connects it to functional behavior. We develop the finite depth setting here, with further discussion in Appendix H.

Our main contributions are:

1. **Path law framework.** We formulate activation steering as path law alignment, separating the target law, intervention family, and discrepancy. We characterize marginal alignment's ambiguity and use Kolmogorov's extension theorem to identify the joint distributions that determine the target law.
2. **Theoretical guarantees.** We prove that, under stated conditions, zero finite depth MMD loss identifies the reference path law, and vanishing loss is equivalent to weak convergence. For actual trajectories that include a shared input, we establish conditions under which sufficiently accurate matching improves functional agreement over exact marginal matching with positive error. We also derive conditions under which marginal MMD recovers the difference-in-means vector as the unique optimal shift.
3. **Practical method and validation.** We construct an efficient objective over contiguous layer groups and optimize steering maps on cached activations without additional backbone passes. Experiments on 7B to 31B models demonstrate improved steering effectiveness for rotational, projection, and affine maps and assess capability retention. Exploratory ablation sweeps examine sensitivity to matching scope, kernels, and weights, while additive map experiments approximately recover the difference-in-means direction.

2 A PATH LAW FRAMEWORK FOR ACTIVATION STEERING

2.1 ACTIVATION TRAJECTORIES AND THE STEERING PROBLEM

We consider a neural network with L layers, hidden dimension d, and fixed pretrained parameters ϑ. For simplicity, assume that the input and all layer activations lie in $\mathbb{R}^d$. The network map is given by

$$f_\vartheta = f_\vartheta^{(L)} \circ f_\vartheta^{(L-1)} \circ \cdots \circ f_\vartheta^{(1)}.$$

Let $X^{(0)} \sim \mu^{(0)}$ denote the input, which is not modified by steering. The activation after layer ℓ is

$$X^{(\ell)} = f_\vartheta^{(\ell)}\left(X^{(\ell-1)}\right), \qquad \ell = 1, \ldots, L.$$

We write $\mu^{(\ell)}$ for its marginal distribution, so that $X^{(\ell)} \sim \mu^{(\ell)}$. These activations form the activation trajectory

$$\mathcal{X} = \left(X^{(0)}, X^{(1)}, \ldots, X^{(L)}\right).$$

Following stochastic process notation (Øksendal, 2013), we refer to the distribution over entire activation trajectories as the path law. Throughout the paper, we sometimes use the terms law and distribution interchangeably for both random variables and stochastic processes. For ease of exposition, we defer the precise mathematical definitions of trajectories and path laws until Section 2.2.

The path law retains dependence across layers, including the relationship between inputs and outputs. Steering therefore targets this law rather than only the activation distribution at each layer.

Given source trajectories $\{\mathcal{X}_{s,i}\}_{i=1}^n$ and reference trajectories $\{\mathcal{X}_{r,i}\}_{i=1}^m$, sampled from path laws $\mathcal{D}_s$ and $\mathcal{D}_r$, respectively, activation steering seeks a measurable map $\mathcal{S}_\theta$ with parameters $\theta \in \Theta \subseteq \mathbb{R}^p$,

$$\mathcal{S}_\theta : (\mathbb{R}^d)^{L+1} \to (\mathbb{R}^d)^{L+1}$$

Given a source trajectory $\mathcal{X}_s$, we write the steered trajectory as

$$\widetilde{\mathcal{X}}_\theta := \mathcal{S}_\theta(\mathcal{X}_s) = \left(X_s^{(0)}, \widetilde{X}_\theta^{(1)}, \ldots, \widetilde{X}_\theta^{(L)}\right),$$

where $\widetilde{X}_\theta^{(0)} = X_s^{(0)}$ leaves the input unchanged. For $\mathcal{X}_s \sim \mathcal{D}_s$, the transformed trajectory has law $\mathcal{D}_\theta = \mathrm{Law}(\mathcal{S}_\theta(\mathcal{X}_s))$. Steering then seeks parameters θ that align this law with the reference law $\mathcal{D}_r$. Actual intervened trajectories and transformed cached activations generally have different laws; Appendix C.5 formalizes this distinction.

This raises a central question: If steering perfectly matches the reference activation distribution at every layer, does it also match the reference path law? As we show next, even perfect layerwise alignment can leave fundamentally different activation trajectories indistinguishable.

2.2 Why marginal matching is insufficient

To examine this question, let $\mu_s^{(\ell)}$ and $\mu_r^{(\ell)}$ denote the source and reference marginal distributions at layer ℓ. Most existing activation steering formulations choose the parameters θ_ℓ of maps $\mathcal{S}_{\theta_\ell}^{(\ell)} : \mathbb{R}^d \to \mathbb{R}^d$ either through heuristics or training, aiming to match these marginals at each selected layer $\ell \in \{1, \dots, L\}$:

$$\mathcal{S}_{\theta_\ell}^{(\ell)}\left(X_s^{(\ell)}\right) \sim \mu_r^{(\ell)}, \qquad X_s^{(\ell)} \sim \mu_s^{(\ell)}.$$

To place finite depth networks within a general stochastic process formulation, we view activations across layers as an *activation process*. We represent this process using a continuous depth index $t \in (0, T]$, holding each layer's activation constant over its corresponding interval:

$$X(t) = X^{(\ell)}, \qquad \text{where} \quad \frac{\ell T}{L+1} < t \leq \frac{(\ell+1)T}{L+1}, \qquad \ell = 0, 1, \dots, L.$$

We now define the trajectory of a stochastic process and its path law.

Definition 2.1 (Trajectory and path law (Øksendal, 2013))**.** Let $\{X(t)\}_{t\in(0,T]}$ be an $\mathbb{R}^d$-valued stochastic process on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For each $\omega \in \Omega$, the function

$$\mathcal{X}_\omega : (0, T] \to \mathbb{R}^d, \qquad \mathcal{X}_\omega(t) = X(t, \omega),$$

is called a trajectory, or path, of the process. The path law $\mathcal{D}$ is the probability distribution of these trajectories, defined by

$$\mathcal{D}(A) = \mathbb{P}\big(\{\omega \in \Omega : \mathcal{X}_\omega \in A\}\big)$$

for every measurable set of paths A.

Remark 2.2. A trajectory records activations across depths for one outcome, while the path law describes the distribution of these trajectories and captures dependencies across depths. Appendix A gives the full measure theoretic definitions.

Consider two processes $\{X_s(t)\}_{t\in(0,T]}$ and $\{X_r(t)\}_{t\in(0,T]}$, with marginal distributions $\mu_s^{(t)}$ and $\mu_r^{(t)}$, respectively. We ask whether

$$\mu_s^{(t)} = \mu_r^{(t)} \qquad \text{for all } t \in (0, T],$$

implies equality of their path laws?

The answer to this question is negative, since we can construct two processes that share the same marginal distributions at every depth while possessing very different path laws.

Figure 1 illustrates this non-uniqueness. We fix a sequence of Gaussian-mixture marginals before constructing two scalar networks with the same input. Network A preserves percentile ranks between layers, whereas network B folds these ranks while preserving their uniform distribution. Both therefore realize the fixed marginals, which vary with depth, but have distinct path laws. Appendix A (Figure A1) gives a simpler example: one process leaves activations unchanged, while the other reverses their signs. Both have the same layer distributions but different path laws when starting from $\mathcal{N}(0, 1)$.

An objective based solely on layerwise marginals cannot distinguish these networks, despite their different activation dynamics. This motivates matching the evolution of activations across layers.

We therefore formulate steering as matching the steered path law $\mathcal{D}_\theta$ to the reference law $\mathcal{D}_r$:

$$\min_{\theta\in\Theta} \mathfrak{D}(\mathcal{D}_\theta, \mathcal{D}_r), \tag{1}$$

where $\mathfrak{D}$ measures the discrepancy between these laws. The reference law specifies the target, the steering map determines attainable laws, and the discrepancy defines the fitting criterion. We develop this framework for finite depth neural networks; Appendix H.1 illustrates a simple extension of this formulation to other stochastic models.

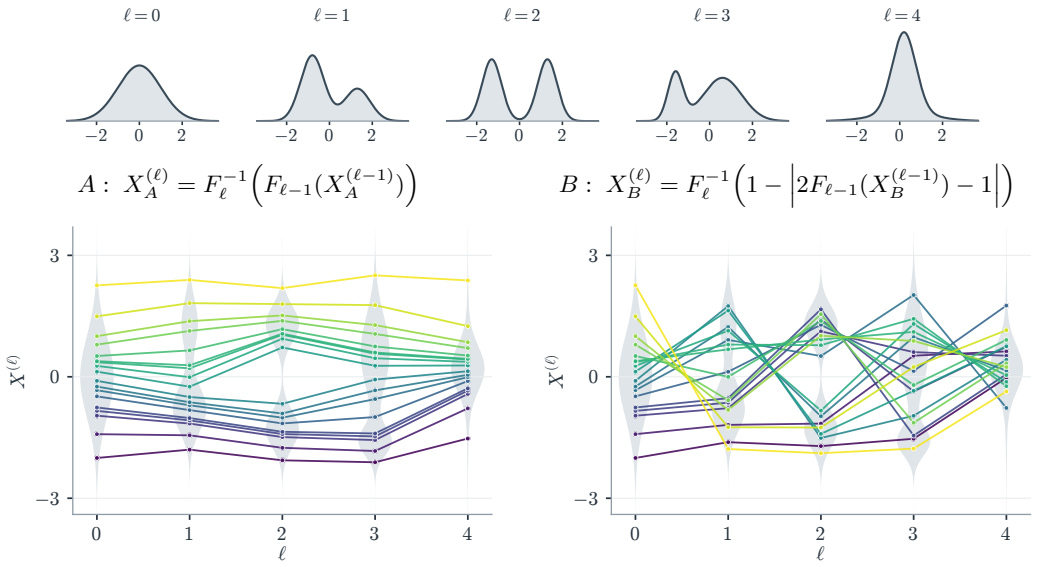


Figure 1: Monotone transport and folding share the Gaussian mixture marginals as shown above but produce different trajectories below. Here, F_ℓ and F_ℓ^{-1} are the CDF and quantile function of the layer ℓ distribution. Matching colors denote the same inputs; gray shading shows the shared densities.

3 From Path Law Alignment to Kolmogorov Loss

Matching activation path laws clearly requires more than aligning individual layer distributions, but it is not yet clear how to match these laws. Fortunately, the celebrated Kolmogorov extension theorem (Kolmogoroff, 1933) provides the starting point: it uniquely characterizes a path law through its finite dimensional distributions. For finite depth networks, we build on this characterization to construct an objective using joint MMD over contiguous groups of layers.

3.1 Path law identification using Kolmogorov extension theorem

We state Kolmogorov's extension theorem following the modern formulation in Øksendal (2013).

Theorem 3.1 (Kolmogorov extension theorem (Øksendal, 2013))**.** *For every $k \geq 1$ and distinct indices $t_1, \ldots, t_k \in (0, T]$, let $\mu^{(t_1,\ldots,t_k)}$ be a Borel probability measure on $(\mathbb{R}^d)^k$. Suppose this family satisfies the following consistency conditions:*

1. *Permutation consistency: for every permutation π of $\{1, \ldots, k\}$ and Borel sets $A_1, \ldots, A_k \subseteq \mathbb{R}^d$,*
$$\mu^{(t_{\pi(1)},\ldots,t_{\pi(k)})}(A_{\pi(1)} \times \cdots \times A_{\pi(k)}) = \mu^{(t_1,\ldots,t_k)}(A_1 \times \cdots \times A_k).$$
2. *Marginal consistency: for every additional distinct index $t_{k+1} \in (0, T] \setminus \{t_1, \ldots, t_k\}$,*
$$\mu^{(t_1,\ldots,t_k,t_{k+1})}(A_1 \times \cdots \times A_k \times \mathbb{R}^d) = \mu^{(t_1,\ldots,t_k)}(A_1 \times \cdots \times A_k).$$

Then there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and an $\mathbb{R}^d$-valued stochastic process $\{X(t)\}_{t\in(0,T]}$ such that
$$\big(X(t_1), \ldots, X(t_k)\big) \sim \mu^{(t_1,\ldots,t_k)}$$
for every finite collection of distinct indices. Its path law $\mathcal{D}$ is uniquely determined on $(\mathbb{R}^d)^{(0,T]}$ equipped with the product σ-algebra.

Remark 3.2. For a neural network, the finite joint distributions of activations at selected layers trivially satisfy these consistency conditions: reordering the layers simply reorders the coordinates, and marginalizing out a layer gives the distribution of the remaining layers. Consequently, two activation processes have the same path law if and only if their joint distributions agree for every collection of depth indices.

3.2 THE KOLMOGOROV LOSS FOR FINITE DEPTH NETWORKS

Inspired by the Kolmogorov extension theorem (Theorem 3.1), we aim to match joint activation distributions across depth. For this we use maximum mean discrepancy (MMD), which compares kernel mean embeddings and can be estimated directly from samples (Gretton et al., 2012). We choose MMD for this purpose because it has been widely used for joint distribution matching in machine learning (Long et al., 2017; Gao & Huang, 2018; Jin et al., 2020).

Remark 3.3. The path law formulation specifies for us which distributions to match, while leaving the choice of discrepancy metric open. MMD is only one possible choice; alternatives include optimal transport based discrepancies, such as Sinkhorn divergences (Genevay et al., 2018), and f-divergences estimated through variational objectives (Nowozin et al., 2016). These alternatives could also be applied to joint activation distributions, with their respective estimation and optimization procedures. From here on, we focus on MMD and develop an efficient implementation and theoretical guarantees for this choice.

For distinct depths $I = (\ell_1, \dots, \ell_k)$ in $\{0, \dots, L\}$, define

$$z_i^I(\theta) = \big(\widetilde{X}_{\theta,i}^{(\ell_1)}, \dots, \widetilde{X}_{\theta,i}^{(\ell_k)}\big), \qquad w_j^I = \big(X_{r,j}^{(\ell_1)}, \dots, X_{r,j}^{(\ell_k)}\big),$$

with laws μ_θ^I and μ_r^I on $\mathcal{Z}_I = (\mathbb{R}^d)^k$. Here, $\widetilde{\mathcal{X}}_{\theta,i} = \mathcal{S}_\theta(\mathcal{X}_{s,i})$, and depth 0 is the unchanged input. Let k_ℓ be a real valued, bounded, continuous positive definite kernel with reproducing kernel Hilbert space $\mathcal{H}_{k_\ell}$. Define

$$K_I(z, w) = \prod_{a=1}^{k} k_{\ell_a}(z_a, w_a), \qquad \mathcal{L}_I(\theta) = \mathrm{MMD}^2_{K_I}(\mu_\theta^I, \mu_r^I).$$

If each k_ℓ is characteristic and translation invariant, K_I is characteristic (Szabó & Sriperumbudur, 2018, Theorem 4), so $\mathcal{L}_I(\theta) = 0$ exactly when the two laws agree. Write $\widehat{\mathcal{L}}_I$ for the squared MMD between their empirical distributions; Appendix B.1 gives its sample formula and the kernel construction.

An objective covering every nonempty collection of layers would contain $2^{L+1} - 1$ terms, which is expensive for deep networks. One can notice that matching only the full trajectory identifies its law, but limited steering data can leave distinct joint distributions in the high dimensional trajectory space statistically indistinguishable under MMD loss. We therefore combine the full trajectory with smaller groups of layers to balance computational and statistical efficiency.

Since each layer processes the preceding layer's activations, contiguous groups are a natural choice. We include $I_{a:b} = (a, \dots, b)$ for $0 \le a \le b \le L$: individual layers, adjacent pairs, and longer groups up to the full trajectory. These $\frac{(L+1)(L+2)}{2}$ groups define the Kolmogorov loss

$$\widehat{\mathcal{L}}_{\mathrm{Kol}}(\theta) = \sum_{0 \le a \le b \le L} w_{a,b} \widehat{\mathcal{L}}_{I_{a:b}}(\theta), \qquad w_{a,b} \ge 0, \qquad \sum_{0 \le a \le b \le L} w_{a,b} = 1. \tag{2}$$

The weights balance shorter and longer groups and remain fixed while all steering parameters are optimized jointly.

For full trajectories $z, z' \in (\mathbb{R}^d)^{L+1}$, the aggregate kernel is

$$K_{\mathrm{Kol}}(z, z') = \sum_{0 \le a \le b \le L} w_{a,b} K_{I_{a:b}}(z_{a:b}, z'_{a:b}), \tag{3}$$

where $z_{a:b} = (z_a, \dots, z_b)$. Replacing empirical losses by population losses gives $\mathcal{L}_{\mathrm{Kol}}(\theta) = \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\mathcal{D}_\theta, \mathcal{D}_r)$. Theorem 4.3 gives conditions under which this loss identifies the full path law.

Contiguous groups allow layerwise kernel values or distances to be reused through cumulative products or sums. Appendix B provides implementation of these formulas and the detailed construction of kernels along with their bandwidths.

4 THEORETICAL ANALYSIS

The discussion in Section 2.2 highlights a fundamental limitation of matching distributions at individual layers: the marginals do not uniquely determine how activations across layers are coupled along a trajectory. We now examine this non-uniqueness more formally.

Proposition 4.1. *Let U and V be stochastic processes on a common probability space with identical marginals but distinct path laws $\mathcal{D}_U$ and $\mathcal{D}_V$. For each $\lambda \in [0,1]$, define*

$$W_\lambda(t) = (1 - B_\lambda)U(t) + B_\lambda V(t), \qquad t \in (0, T],$$

where $B_\lambda \sim \text{Bernoulli}(\lambda)$ is independent of both processes and selects one process for the entire trajectory. The resulting path laws

$$\mathcal{D}_\lambda = (1-\lambda)\mathcal{D}_U + \lambda \mathcal{D}_V, \qquad \lambda \in [0,1],$$

form an uncountable family of distinct path laws sharing the same marginals.

Remark 4.2. Proposition 4.1 shows that any two stochastic processes with identical marginals but distinct path laws generate an uncountable family of distinct path laws sharing these marginals. Figure 1 provides such a pair of stochastic processes from which to construct this family. It is worth noting that other path laws can share the same marginals without belonging to this family, so this construction is by no means exhaustive.

An important question is whether the Kolmogorov loss distinguishes the path laws that marginal matching leaves indistinguishable? The following theorem shows that, under the stated kernel and weighting conditions, the loss is zero exactly when the steered and reference path laws agree.

Theorem 4.3 (Full trajectory matching through consecutive groups)**.** *Let K_{Kol} be the kernel in Eq. 3 on $\mathcal{Z} = (\mathbb{R}^d)^{L+1}$, with fixed real valued, bounded, continuous, positive definite depth kernels k_ℓ, $\ell = 0, \dots, L$, and fixed nonnegative weights satisfying $\sum_{0 \le a \le b \le L} w_{a,b} = 1$ and $w_{0,L} > 0$. Assume that each RKHS $\mathcal{H}_{k_\ell}$ is contained in and uniformly dense in $C_0(\mathbb{R}^d)$, the space of continuous functions vanishing at infinity.*

Then K_{Kol} is bounded, continuous, and characteristic. For every $\theta \in \Theta$,

$$\mathcal{L}_{\text{Kol}}(\theta) = \text{MMD}^2_{K_{\text{Kol}}}(\mathcal{D}_\theta, \mathcal{D}_r). \tag{4}$$

In particular, $\mathcal{L}_{\text{Kol}}(\theta) = 0$ if and only if $\mathcal{D}_\theta = \mathcal{D}_r$.

Moreover, writing $I_{0:L} = (0, \dots, L)$, for every sequence $(\theta_n) \subseteq \Theta$ and fixed reference law $\mathcal{D}_r$,

$$\mathcal{L}_{\text{Kol}}(\theta_n) \longrightarrow 0 \quad \Longleftrightarrow \quad \mathcal{L}_{I_{0:L}}(\theta_n) \longrightarrow 0 \quad \Longleftrightarrow \quad \mathcal{D}_{\theta_n} \Rightarrow \mathcal{D}_r, \tag{5}$$

where $\Rightarrow$ denotes weak convergence.

Remark 4.4. The identity in Equation 4 also holds for empirical probability measures, so the empirical objective is exactly a squared MMD between the empirical trajectory laws. The density assumption is satisfied by Gaussian, Laplace, and rational quadratic kernels. Applying the weak convergence characterization of Simon-Gabriel et al. (2023, Theorem 9) to the full trajectory kernel, together with its strictly positive weight and the bounded continuity of the remaining group kernels, yields the stated convergence equivalence. A complete proof is provided in Appendix C.2.

Theorem 4.3 shows that the Kolmogorov loss identifies the full path law. We next ask whether matching this law brings the model's outputs closer to the reference outputs on the same inputs.

Theorem 4.5 (Functional improvement over marginal matching)**.** *Let the actual steered and reference trajectories share input $X^{(0)} \sim \mu^{(0)}$ and be deterministic given that input. Define their output functions by $h_\theta(X^{(0)}) = \widetilde{X}_\theta^{(L)}$ and $f(X^{(0)}) = X_r^{(L)}$, respectively. Assume their path laws, for all $\theta \in \Theta$, lie on a common compact set $\mathcal{Z} \subseteq (\mathbb{R}^d)^{L+1}$, f is continuous on the input projection of $\mathcal{Z}$, and K_{Kol} is continuous and characteristic. Let $\mathcal{L}_{\text{Kol}}$ be the population loss between these path laws and define the output error $R(\theta) = \mathbb{E}_{X^{(0)} \sim \mu^{(0)}}[\|h_\theta(X^{(0)}) - f(X^{(0)})\|^2]$.*

For any $\theta^{\text{marg}} \in \Theta$ that matches every reference layer marginal but has $R(\theta^{\text{marg}}) > 0$, there exists $\tau > 0$ such that, for all $\theta \in \Theta$,

$$\mathcal{L}_{\text{Kol}}(\theta) < \tau \quad \Longrightarrow \quad R(\theta) < R(\theta^{\text{marg}}).$$

The threshold construction and the guarantee allowing approximation and optimization errors are given in Appendix C.3.

Remark 4.6. The matched law in Theorem 4.5 includes the shared input and actual intervened activations, retaining their correspondence. This result does not require a particular steering map parameterization, provided the map family can achieve sufficiently small population loss. Appendix C.5 gives a conditional bound relating cached and actual trajectory discrepancies and explains why matching internal activations alone does not give this functional guarantee.

The preceding theorem requires a steering family that can sufficiently reduce the matching objective. However, in practice simple additive maps can perform well despite their limited expressiveness. We therefore examine the minimizer of the MMD objective when the map is additive. The following theorem proves that the difference-in-means vector uniquely minimizes this objective under certain distributional conditions.

Theorem 4.7 (Exact recovery of the difference in means vector)**.** *Let $X_s, X_r \in \mathbb{R}^d$ have finite means and centered distributions whose characteristic functions are real and strictly positive at every frequency. For any real valued, bounded, continuous, translation invariant characteristic kernel k, the marginal MMD between $\mathrm{Law}(X_s + b)$ and $\mathrm{Law}(X_r)$ is uniquely minimized by $b^\star = \mathbb{E}[X_r] - \mathbb{E}[X_s]$.*

Remark 4.8. Theorem 4.7 covers widely used kernels such as Gaussian, rational quadratic, and Laplace, while allowing the source and reference distributions to differ in shape. In practice, recovering the difference in means vector is somewhat expected: additive maps can shift means but cannot change higher order centered moments or inter layer dependence among the activations they directly translate. Later, in Section 5.3, we empirically show that MMD optimization from random initialization approximately recovers this direction on real activation data, despite more complex activation distributions and finite sample estimation noise.

5 Experimental Results

In this section, we evaluate our framework on jailbreak and truthfulness tasks by optimizing existing steering maps with the Kolmogorov loss and comparing them with their baselines. We apply the maps to cached activations without additional forward or backward passes through the pretrained model. This allows parallel map evaluation across layers while the joint loss matches cross layer dependencies. Because downstream activations are not recomputed after earlier interventions, this objective serves as a surrogate for matching actual intervened trajectories. Theorem 4.5 concerns those actual trajectories under its stated assumptions; the effectiveness of cached fitting is evaluated empirically. Appendix C.5 bounds the cached approximation under stability assumptions, and Appendix H discusses its computational benefits.

5.1 Jailbreak

This task aims to bypass a model's refusal mechanism and steer it to answer harmful requests. The source and reference distributions correspond to activation trajectories induced by harmful and harmless requests, respectively. We evaluate three models: Qwen3.5-27B, GPT-OSS-20B, and Gemma-4-31B-IT. We consider two steering methods: Adaptive Angular Steering (AAS) Vu & Nguyen (2025) and Directional Ablation (DirAbl) Arditi et al. (2024), which use rotation and projection maps, respectively. For each method, Kolmogorov steering replaces heuristic parameter selection with parameters learned by minimizing the proposed MMD loss in Eq. 2. We measure jailbreak effectiveness using HarmBench-classified attack success rates (ASR) (Mazeika et al., 2024) and retention of general model capabilities using TinyBench scores (Maia Polo et al., 2024). Further details on the experimental setup, map selection results, and fitting performance are provided in Appendix D.

Table 1 compares the unsteered models, the original steering methods, and their Kolmogorov variants. Kolmogorov steering improves ASR in five of the six method-model comparisons, with substantial gains in cases where the original constructions perform poorly. Specifically, original AAS reduces TinyBench performance on GPT-OSS-20B by 7.59 percentage points, while original DirAbl achieves an attack success rate of only 8.65% on Qwen3.5-27B. In both cases, Kolmogorov steering substantially improves the affected metric.

Table 1: HarmBench ASR and TinyBench average scores (%; higher is better) for AAS, DirAbl, and their Kolmogorov variants. Parentheses show TinyBench changes from the unsteered baseline in percentage points. Bold marks the higher ASR in each method pair.

	Qwen3.5-27B		GPT-OSS-20B		Gemma-4-31B-IT	
Method	HarmBench ASR ↑	TinyBench Avg. ↑	HarmBench ASR ↑	TinyBench Avg. ↑	HarmBench ASR ↑	TinyBench Avg. ↑
Baseline (no steer)	0.00	72.78 (+0.00)	0.96	70.83 (+0.00)	0.96	72.65 (+0.00)
AAS	**89.42**	71.98 (−0.80)	91.35	63.24 (−7.59)	65.38	71.31 (−1.34)
Kolmogorov AAS	88.46	72.38 (−0.40)	**96.15**	70.84 (+0.01)	**82.69**	71.89 (−0.77)
DirAbl	8.65	72.27 (−0.51)	78.85	66.39 (−4.44)	83.65	72.77 (+0.11)
Kolmogorov DirAbl	**73.08**	71.96 (−0.82)	**93.27**	69.41 (−1.42)	**90.38**	71.51 (−1.15)

Optimizing the AAS and DirAbl maps from cached activations takes 42-80 seconds, excluding activation extraction, configuration selection, and downstream evaluation. The learned maps are then reused throughout inference. Further timing details are provided in Appendix D.3.

5.2 TRUTHFULNESS

This task aims to improve the truthfulness of open ended responses in three base language models: Llama-3.1-8B, Qwen2.5-7B, and Mistral-7B. We use a two fold train/test protocol and intervene at a single selected layer. Steering maps are optimized to match cached activations induced by incorrect reference answers to those induced by correct reference answers. We focus on affine steering maps, $S_{A,b}(x) = Ax + b$, and consider two established methods: Linear-AcT (Rodríguez et al., 2025a) and MiMiC (Singh et al., 2024). We pair Linear-AcT with Kolmogorov Diagonal, both of which use a diagonal A. We pair MiMiC, which uses a full matrix A, with Kolmogorov Low Rank, which uses an identity map with a rank-2 learned correction. Both Kolmogorov variants cache activations from all decoder layers and learn (A, b) by minimizing the proposed MMD loss over the full cached trajectory, with the map applied only at the selected layer. All methods are evaluated on the TruthfulQA generation benchmark (Lin et al., 2022) to assess truthfulness and informativeness. The primary metric, $T \times I$, is the percentage of responses judged jointly truthful and informative. Further details on the experimental setup, map fitting, and configuration selection are provided in Appendix E.

Table 2: Truthfulness and informativeness scores (%; higher is better). Truth and Info measure truthful and informative responses, respectively; $T \times I$ measures responses judged both. CAA uses a strength selected on validation data; Kolmogorov maps, Linear-AcT, and MiMiC use full strength. Bold marks the higher $T \times I$ in each affine method pair.

	Llama-3.1-8B			Qwen2.5-7B			Mistral-7B		
Method	T×I ↑	Truth ↑	Info ↑	T×I ↑	Truth ↑	Info ↑	T×I ↑	Truth ↑	Info ↑
Baseline (no steer)	41.7	43.2	98.2	64.1	77.8	85.6	38.7	41.0	97.1
CAA (DIM)	60.7	67.2	93.3	65.0	79.2	85.1	61.1	67.8	93.3
Linear-AcT	48.3	50.2	97.9	66.8	79.6	86.9	44.9	47.9	96.8
Kolmogorov Diagonal	**50.9**	60.3	89.8	**72.0**	82.4	89.4	**52.6**	74.7	77.5
MiMiC	51.8	59.0	92.2	61.6	87.3	74.1	43.5	48.5	94.9
Kolmogorov Low Rank	**59.0**	64.1	94.6	**69.4**	83.2	85.8	**60.0**	69.8	90.0

Table 2 compares Linear-AcT with Kolmogorov Diagonal and MiMiC with Kolmogorov Low Rank, with CAA included as an additive baseline. Kolmogorov Diagonal improves both truthfulness and $T \times I$ over Linear-AcT across all three models, including a 26.8-percentage-point gain in truthfulness on Mistral-7B. Kolmogorov Low Rank also improves $T \times I$ over MiMiC across all three models. It increases truthfulness on Llama-3.1-8B and Mistral-7B, while on Qwen2.5-7B, a modest

reduction in truthfulness is accompanied by a larger gain in informativeness. Because MiMiC and Kolmogorov Low Rank use different parameterizations, their comparison does not isolate the effect of the matching loss.

5.3 Assessing the optimality of CAA's DIM vector

Theorem 4.7 establishes the optimality of CAA's DIM vector (Rimsky et al., 2024) for the MMD objective across several distribution families. Here, we examine whether this prediction carries over to model activations. Alongside cosine similarity, we report the absolute relative norm difference. Consistent with Theorem 4.7, MMD optimization from random initialization recovers vectors with high cosine similarity to DIM across most evaluated settings. Relative norm differences remain below 2% except on GPT-OSS-20B, where the difference is 13.27% (Tables 3 and 4). Taken together, these findings show that optimizing the proposed matching objective approximately recovers the DIM direction in these settings. Thus, the room for improvement may therefore lie less in finding a better translation vector than in developing more expressive steering maps. We provide the experimental details and performance of both DIM vector and Kolmogorov learnt vector in Appendix F.

Table 3: Jailbreak vector agreement between DIM and Kolmogorov CAA fitted from random initialization.

Model	Cosine	Norm diff. (%)
Qwen3.5-27B	0.999813	0.98
GPT-OSS-20B	0.976119	13.27
Gemma-4-31B-IT	0.997130	1.85

Table 4: Truthfulness vector agreement between DIM and randomly initialized Kolmogorov CAA, averaged over two outer folds.

Model	Cosine	Norm diff. (%)
Llama-3.1-8B	0.996484	0.91
Qwen2.5-7B	0.998062	1.90
Mistral-7B	0.996299	0.97

5.4 Ablation Study of Matching Scope, Kernels, and Weights

On Qwen3.5-9B and Llama-3.1-8B-Instruct, we compare marginal matching, full trajectory matching, and matching over all contiguous layer groups to assess steering effectiveness and capability retention. We further examine how the choice of kernel and interval weights affects these outcomes. For the kernel comparison, we use the Gaussian and rational quadratic (RQ) kernels defined in Appendix B, with RQ shape parameter $\alpha = 1$. For interval weighting, we compare $w_{a,b} \propto 1$ and $w_{a,b} \propto (b - a + 1)^{-1}$, referred to as uniform and short weighting, respectively. The latter assigns greater weight to shorter groups of consecutive layers. All weights are normalized to sum to one.

Table A4 in Appendix G reports the result with the same task settings as Section 5.1. From this table, uniformly weighted all group matching with Gaussian kernels exceeds marginal matching by 4.81 and 1.92 ASR points on Qwen and Llama, respectively, alongside TinyBench gains of 0.82 and 0.28 points.

6 Conclusion

In this paper, we provide a theoretical framework for activation steering based on matching activation path laws. On the theoretical side, we characterize the ambiguity of marginal alignment and use Kolmogorov's extension theorem to identify the joint distributions that determine the target law. For finite depth networks, we establish conditions under which the Kolmogorov loss identifies this law and sufficiently accurate matching of actual trajectories with shared inputs improves functional agreement. We also derive conditions under which the difference-in-means vector is the unique optimal additive shift. On the empirical side, we construct Kolmogorov loss as an efficient objective that reuses kernel evaluations and enables parallel fitting on cached activations without additional backbone passes. Experiments on jailbreak and truthfulness across language models with 7B to 31B parameters demonstrate its utility across several steering map families. A limitation is that the functional improvement theoretical result assumes actual trajectories with shared inputs, whereas the experiments match cached internal activations from different prompts. We provide a conditional bound for the cached approximation, but its error and the missing input correspondence remain uncontrolled in these experiments. Extending functional guarantees to this setting remains an interesting direction to expand upon in future work.

AI USE STATEMENT

In this work, we used generative AI tools for exploring related literature. The core research ideas, theoretical formulation, mathematical claims, and experimental design were developed by the authors.

Additionally, we used generative AI tools for editing software code and assisting with the writing and editing of the manuscript. We have reviewed all AI-assisted work. In particular, suggestions obtained during AI-assisted exploration were critically evaluated by the authors, all theoretical arguments and proofs in the final manuscript were carefully checked and verified by the authors, relevant references were checked against the original sources, and the experimental implementation and reported results were inspected by the authors.

We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies interventions that can change both desired and undesired model behaviors. In particular, jailbreak steering can weaken refusal mechanisms and enable harmful responses, creating potential misuse risks. Our benchmark evaluations examine these mechanisms and their effects on general capabilities. Automated safety and truthfulness scores are imperfect proxies for human judgments and do not establish safe or reliable deployment.

REPRODUCIBILITY STATEMENT

The assumptions and proofs are given in Appendix C, including the conditional analysis of cached fitting in Appendix C.5. Appendix B specifies the kernels, bandwidth selection, and cumulative evaluation. Data splits, activation sites, map parameterizations, optimization, configuration selection, and evaluation procedures are described in Appendices D, E, and F. The exploratory ablation selection protocol is stated in Appendix G. Reported scores are point estimates from the specified fits and splits.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 136037–136083, 2024. doi: 10.52202/079017-4322. URL https://papers.nips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2004. doi: 10.1007/978-1-4419-9096-9. URL https://link.springer.com/book/10.1007/978-1-4419-9096-9.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Hongchang Gao and Heng Huang. Joint generative moment-matching network for learning structural latent code. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 2121–2127, 2018. doi: 10.24963/ijcai.2018/293. URL https://www.ijcai.org/proceedings/2018/293.

Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1608–1617, 2018.

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

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023b. doi: 10.18653/v1/2023.emnlp-main.825.

Xiaona Jin, Xiaowei Yang, Bo Fu, and Sentao Chen. Joint distribution matching embedding for unsupervised domain adaptation. *Neurocomputing*, 412:115–128, 2020. doi: 10.1016/j.neucom.2020.05.098. URL https://www.sciencedirect.com/science/article/abs/pii/S0925231220309802.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A. Kolmogoroff. *Grundbegriffe der Wahrscheinlichkeitsrechnung*, volume 2 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 1. Folge*. Springer, Berlin, Heidelberg, 1933. doi: 10.1007/978-3-642-49888-6. URL https://link.springer.com/book/10.1007/978-3-642-49888-6.

Ho Chung Law, Christopher Yau, and Dino Sejdinovic. Testing and learning on distributions with symmetric noise invariance. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/67d16d00201083a2b118dd5128dd6f59-Abstract.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2208–2217. PMLR, 2017. URL https://proceedings.mlr.press/v70/long17a.html.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20619–20634. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1007. URL https://aclanthology.org/2025.acl-long.1007/.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Bernt Øksendal. *Stochastic differential equations: an introduction with applications*. Universitext. Springer, 5 edition, 2013. doi: 10.1007/978-3-662-03620-4. URL https://link.springer.com/book/10.1007/978-3-662-03620-4.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

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. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Pau Rodríguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=l2zFn6TIQi.

Pau Rodríguez, Michal Klein, Eleonora Gualdoni, Valentino Maiorca, Arno Blaas, Luca Zappella, Marco Cuturi, and Xavier Suau. LinEAS: End-to-end learning of activation steering with a distributional loss. In *Advances in Neural Information Processing Systems*, volume 38, pp. 82356–82398, 2025b. doi: 10.52202/085713-2756. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/767ef009ca91f00d1e256bc6f3fc7969-Abstract-Conference.html.

Carl-Johann Simon-Gabriel, Alessandro Barp, Bernhard Schölkopf, and Lester Mackey. Metrizing weak convergence with maximum mean discrepancies. *Journal of Machine Learning Research*, 24(184):1–20, 2023. URL https://jmlr.org/papers/v24/21-0599.html.

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45663–45680. PMLR, 2024. URL https://proceedings.mlr.press/v235/singh24d.html.

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(50):1517–1561, 2010. URL https://jmlr.org/papers/v11/sriperumbudur10a.html.

Zoltán Szabó and Bharath K. Sriperumbudur. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18(233):1–29, 2018. URL https://jmlr.org/papers/v18/17-492.html.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, et al. Advancing mathematics research with AI-driven formal proof search. *arXiv preprint arXiv:2605.22763*, 2026.

Minh Hieu Vu and Tan Nguyen. Angular steering: Behavior control via rotation in activation space. *Advances in Neural Information Processing Systems*, 38:121653–121690, 2025.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek Abdelzaher, Yejin Choi, Manling Li, and Huajie Shao. Odesteer: A unified ode-based steering framework for llm alignment. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 69961–69988, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/72136a6372df75ddc3b1cd7f27c63712-Paper-Conference.pdf.

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

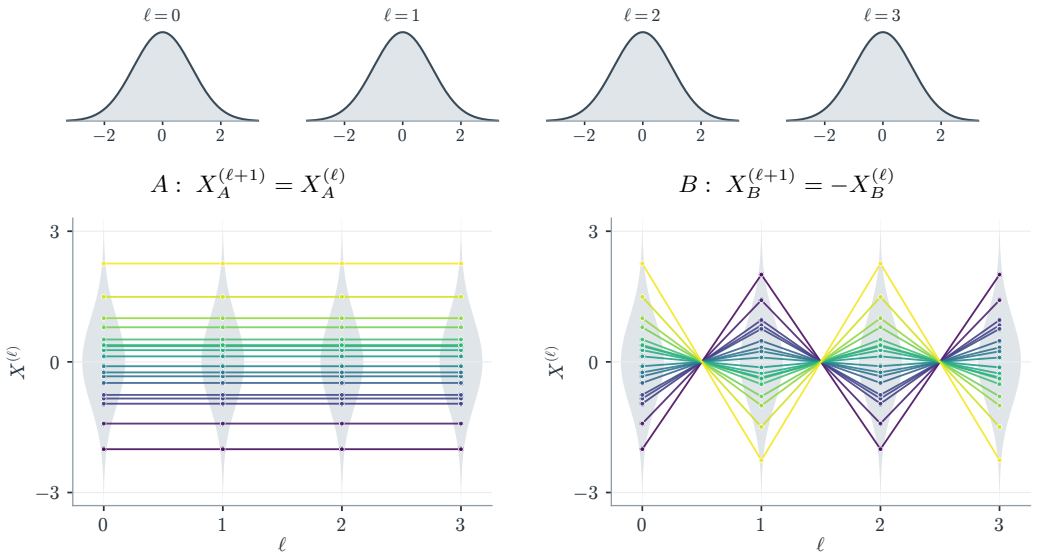


Figure A1: Identity and sign flip. Both networks receive the same input $X \sim \mathcal{N}(0, 1)$ and apply the indicated map at every layer. The upper panels show their common marginal densities; the lower panels show trajectories for the same inputs. Colors identify inputs by their initial percentile. Gray profiles show the shared densities. The terminal maps are x and $-x$.

A Trajectories and path laws

We give the definition for trajectory and path law using the standard stochastic process formulation of Øksendal (2013).

Definition A.1 (Trajectory and path law)**.** Let $\{X(t)\}_{t\in(0,T]}$ be an $\mathbb{R}^d$-valued stochastic process on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space, $\mathcal{F}$ is a σ-algebra of events, and $\mathbb{P}$ is a probability measure. Each $X(t)$ is measurable from $(\Omega, \mathcal{F})$ to $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mathcal{B}(\mathbb{R}^d)$ is the Borel σ-algebra generated by the open subsets of $\mathbb{R}^d$. For each $\omega \in \Omega$, the function

$$\mathcal{X}_\omega : (0, T] \to \mathbb{R}^d, \qquad \mathcal{X}_\omega(t) = X(t, \omega),$$

is called a trajectory, or path, of the process.

Equip the path space $\mathcal{E} = (\mathbb{R}^d)^{(0,T]}$ with the product σ-algebra

$$\mathcal{B}_{\text{path}} = \sigma\big(\{\{x \in \mathcal{E} : x(t) \in B\} : t \in (0, T],\ B \in \mathcal{B}(\mathbb{R}^d)\}\big)\,.$$

The map $\omega \mapsto \mathcal{X}_\omega$ is measurable because each coordinate $X(t)$ is measurable. Its probability distribution is called the path law $\mathcal{D}$, defined by

$$\mathcal{D}(A) = \mathbb{P}\big(\{\omega \in \Omega : \mathcal{X}_\omega \in A\}\big), \qquad A \in \mathcal{B}_{\text{path}}.$$

Remark A.2. A stochastic process admits two complementary interpretations. At a fixed time t, the map $\omega \mapsto X(t, \omega)$ is a random variable. For a fixed outcome ω, the map $t \mapsto X(t, \omega)$ describes an entire realization of the process, such as the motion of a particle over time. This associates each outcome in Ω with an element of the path space $\mathcal{E}$. The path law transfers the probabilities on Ω to measurable sets of paths, allowing the process to be described on $(\mathcal{E}, \mathcal{B}_{\text{path}}, \mathcal{D})$. This measure theoretic definition covers familiar families of stochastic and deterministic processes, such as solutions to Brownian driven SDEs, Lévy driven SDEs, and ODEs with deterministic or random initial conditions.

B Kernel Construction, Bandwidths, and Cumulative Evaluation

Recorded activation sites. The experimental objective compares activations at the recorded internal sites, without including the unchanged model input. Here, indices $0, \ldots, L$ enumerate the

recorded sites, and the full trajectory term includes all of them. By contrast, Theorem 4.5 includes the shared input to retain the correspondence between inputs and outputs. Because steering leaves the input unchanged, exact matching of these input inclusive path laws requires the source and reference input distributions to agree.

Fix depth scales $s_\ell > 0$ for $\ell = 0, \dots, L$ and interval bandwidths $\tau_{i,j} > 0$ for $0 \le i \le j \le L$, independently of the steering parameters. For $z = (z_0, \dots, z_L)$ and $z' = (z'_0, \dots, z'_L)$, define

$$k_{i:j}(z_{i:j}, z'_{i:j}) = \exp\left[-\frac{1}{2\tau_{i,j}^2(j-i+1)} \sum_{\ell=i}^{j} \frac{\|z_\ell - z'_\ell\|_2^2}{s_\ell^2}\right]. \tag{6}$$

The layer scales are shared across intervals. Each kernel is a Gaussian on an invertibly rescaled Euclidean space and is therefore characteristic (Sriperumbudur et al., 2010). The product of the main text's Gaussian depth kernels with bandwidths s_ℓ is the special case $\tau_{i,j}^2 = 1/(j-i+1)$. The formulas below also allow other fixed interval bandwidths.

Experimental bandwidth selection. The standard fitting configurations use $\tau_{i,j} = 1$. At each activation site, s_ℓ^2 is the median squared Euclidean distance between distinct reference trajectories (Garreau et al., 2017), computed once before optimization and held fixed. With the default pair budget of 4096, the implementation takes the first $\min(N_r, 91)$ reference trajectories and uses all distinct pairs among them, giving 4095 pairs when $N_r \ge 91$. For an interval of length $m = j-i+1$, the effective Gaussian bandwidth at coordinate ℓ is therefore $\sqrt{m}\, s_\ell$. This is an interval dependent product of Gaussian kernels, rather than a product of the same fixed bandwidth depth kernels across all intervals. With finite positive scales, the full interval kernel remains a Gaussian on an invertibly rescaled space, so the identification and weak convergence argument applies using this full interval kernel.

Rational quadratic intervals. For the RQ ablation, the implementation applies a rational quadratic kernel to the normalized distance of the entire interval:

$$k_{i:j}^{\mathrm{RQ}}(z_{i:j}, z'_{i:j}) = \left(1 + \frac{1}{2\alpha\tau_{i,j}^2(j-i+1)} \sum_{\ell=i}^{j} \frac{\|z_\ell - z'_\ell\|_2^2}{s_\ell^2}\right)^{-\alpha}, \qquad \alpha = 1, \quad \tau_{i,j} = 1.$$

This construction uses the same reference scales and cumulative distances as the Gaussian kernel. It is a single RQ kernel on the rescaled interval tuple, not a product of per layer RQ kernels.

Numerically degenerate sites. If a site's median squared reference distance is below the implementation's numerical tolerance, its squared scale is set to infinity, making that site's contribution to every interval distance zero. The jailbreak implementation estimates scales in float64 and the truthfulness implementation in float32, each using a tolerance proportional to machine precision, hidden dimension, and mean squared reference norm. The implemented kernel then compares only the retained coordinates; full path identification requires finite positive scales at every varying coordinate, or that excluded coordinates be fixed and identical across the laws being compared.

Squared distance computation. We compute squared Euclidean distances using the Gram expansion,

$$\widehat{D}(a, b) = \max\{0, \|a\|_2^2 + \|b\|_2^2 - 2a^\top b\}.$$

This avoids storing a d-dimensional difference vector for every trajectory pair at every site. Computing squared distances directly also avoids an unnecessary square root and its singular derivative at zero, which occurs for diagonal pairs. The clamp removes negative distances caused by floating point roundoff.

Cumulative distances. For each trajectory pair, set

$$\delta_\ell = \frac{\|z_\ell - z'_\ell\|_2^2}{s_\ell^2}, \qquad C_{-1} = 0, \qquad C_\ell = \sum_{r=0}^{\ell} \delta_r, \qquad \ell = 0, \dots, L.$$

Writing $\alpha_{i,j} = [2\tau_{i,j}^2(j-i+1)]^{-1}$, we obtain

$$k_{i:j}(z_{i:j}, z'_{i:j}) = \exp[-\alpha_{i,j}(C_j - C_{i-1})]. \tag{7}$$

Computing the distances and cumulative sums costs $O(Ld)$ per pair; each interval then requires constant work. Consequently, all intervals cost $O(Ld + L^2)$, rather than

$$O\left(d\sum_{i=0}^{L}\sum_{j=i}^{L}(j-i+1)\right) = O(dL^3)$$

when their distances are computed independently. For $M = n^2 + m^2 + nm$, where n and m are the source and reference sample counts, exact evaluation of Equation 2 therefore costs $O(M(Ld+L^2))$, excluding map evaluation.

Storage and caching. For a fixed pair, accumulate

$$F(z, z') = \sum_{i=0}^{L}\sum_{j=i}^{L} w_{i,j}\exp[-\alpha_{i,j}(C_j - C_{i-1})]$$

directly into the corresponding sample sum. Storing all interval kernel values would require $O(ML^2)$ scalars. Processing at most B pairs at a time instead requires $O(B(L + d))$ auxiliary storage for distances, cumulative sums, and work arrays for each layer. This excludes resident trajectories, map state, and the fixed interval weights and bandwidths, which require $O(L^2)$ storage when specified individually. No pairwise or interval terms are omitted.

With fixed target data, weights, and bandwidths, the entire target only contribution can be computed once and retained as a scalar. Source–source and source–target distances generally change with θ and are recomputed at each step. After target precomputation, the per step kernel cost is $O((n^2 + nm)(Ld + L^2))$.

Differentiation without retaining interval kernels. The implementation uses a custom backward pass that recomputes interval kernels from the saved layerwise distances rather than retaining every interval kernel value. This avoids the $O(ML^2)$ storage required by automatic differentiation through all interval intermediates. Ordinary automatic differentiation is used for the steering map itself. For a fixed pair,

$$\frac{\partial F}{\partial \delta_\ell} = -\sum_{i\le\ell\le j} w_{i,j}\alpha_{i,j}\exp[-\alpha_{i,j}(C_j - C_{i-1})].$$

These sums are evaluated in $O(L^2)$ time using an array $v_0, \dots, v_{L+1}$, initialized to zero. For each interval, set $h_{i,j} = w_{i,j}\alpha_{i,j}\exp[-\alpha_{i,j}(C_j - C_{i-1})]$ and accumulate

$$v_i \leftarrow v_i - h_{i,j}, \qquad v_{j+1} \leftarrow v_{j+1} + h_{i,j}.$$

A cumulative sum then gives

$$g_\ell = \sum_{r=0}^{\ell} v_r = \frac{\partial F}{\partial \delta_\ell}, \qquad \nabla_{z_\ell} F = \frac{2g_\ell}{s_\ell^2}(z_\ell - z'_\ell), \qquad \nabla_{z'_\ell} F = -\nabla_{z_\ell} F.$$

Thus forward evaluation and activation differentiation both have order $O(M(Ld + L^2))$ arithmetic cost. Pair contributions are accumulated using the sample pair coefficients of the empirical MMD estimator in Appendix B.1, followed by differentiation through the steering map. Since $\widetilde{X}^{(0)}_{\theta,i} = X^{(0)}_{s,i}$ is unchanged, its derivative with respect to θ is zero. Input kernel factors still affect gradients for the other coordinates. The auxiliary bound above excludes the accumulated activation gradients and the map's differentiation state; ordinary automatic differentiation that retains every interval intermediate does not satisfy this bound.

B.1 Joint kernel construction and empirical loss

We give the RKHS construction underlying the product kernel in the main text. For a collection of distinct depths $I = (\ell_1, \dots, \ell_k)$, let $\mathcal{Z}_I = (\mathbb{R}^d)^k$ denote the space of activation tuples. Write $z_i^I(\theta)$ and w_j^I for the steered source and reference tuples, with laws μ_θ^I and μ_r^I, respectively.

For each depth $\ell \in \{0, \dots, L\}$, let k_ℓ be a real valued, bounded, continuous positive definite kernel with Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{k_\ell}$, abbreviated as $\mathcal{H}_\ell$, and feature map $\phi_\ell(x) = k_\ell(x, \cdot)$. The tensor product feature map

$$\Phi_I(z) = \phi_{\ell_1}(z_1) \otimes \cdots \otimes \phi_{\ell_k}(z_k) \quad \text{in} \quad \mathcal{H}_I = \bigotimes_{a=1}^{k} \mathcal{H}_{\ell_a}$$

induces the bounded continuous positive definite kernel

$$K_I(z, w) = \langle \Phi_I(z), \Phi_I(w) \rangle_{\mathcal{H}_I} = \prod_{a=1}^{k} k_{\ell_a}(z_a, w_a).$$

Using this joint feature map, we embed a probability distribution μ on $\mathcal{Z}_I$ as $m_I(\mu) = \mathbb{E}_{Z\sim\mu}[\Phi_I(Z)] \in \mathcal{H}_I$. The MMD associated with K_I is the distance between two such mean embeddings:

$$\mathrm{MMD}_{K_I}(\mu, \nu) = \|m_I(\mu) - m_I(\nu)\|_{\mathcal{H}_I},$$

where μ and ν are probability distributions on $\mathcal{Z}_I$. If each k_ℓ is also characteristic and translation invariant, then K_I is characteristic (Szabó & Sriperumbudur, 2018, Theorem 4), so the mean embedding m_I is injective. The RKHS distance between these embeddings therefore induces a metric, MMD_{K_I}, on the space of Borel probability measures on $\mathcal{Z}_I$. Its square defines the matching objective for the steering parameters θ:

$$\mathcal{L}_I(\theta) = \mathrm{MMD}^2_{K_I}\left(\mu^I_\theta, \mu^I_r\right).$$

Since MMD is a metric under these conditions, $\mathcal{L}_I(\theta)$ is zero if and only if $\mu^I_\theta = \mu^I_r$.

Empirical estimation. For independent samples of n steered source trajectories and m reference trajectories, a finite sample estimate of this squared distance is

$$\widehat{\mathcal{L}}_I(\theta) = \underbrace{\frac{1}{n^2}\sum_{i,i'=1}^{n} K_I(z^I_i(\theta), z^I_{i'}(\theta))}_{\text{(i)}} \underbrace{-2\,\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} K_I(z^I_i(\theta), w^I_j)}_{\text{(ii)}} + \underbrace{\frac{1}{m^2}\sum_{j,j'=1}^{m} K_I(w^I_j, w^I_{j'})}_{\text{(iii)}}.$$

For a fixed collection I, matching is performed by minimizing $\widehat{\mathcal{L}}_I(\theta)$ over $\theta \in \Theta$. Terms (i) and (iii) measure average kernel similarity within the steered source trajectories and reference trajectories, respectively, while (ii) captures similarity between the two sets. With fixed reference trajectories and kernel parameters, term (iii) is independent of θ. Although omitting it would leave the gradients unchanged, the implementation computes it once without gradients and retains the cached value so that the reported loss remains the full empirical squared MMD.

This is the biased empirical squared MMD (Gretton et al., 2012): the sums include diagonal pairs, and the expression equals the squared MMD between the empirical trajectory distributions. The product construction above provides the theoretical starting point. The experimental Gaussian and rational quadratic interval kernels, including their length normalization, are specified earlier in this appendix.

The path law framework leaves the discrepancy open. Alternative choices, including optimal transport based discrepancies and variational divergence objectives, are discussed in Appendix H.

C Proofs for the theoretical analysis

This section presents the proofs of our theoretical results. The arguments use standard tools from measure theoretic probability and functional analysis, including weak convergence of probability measures, reproducing kernel Hilbert spaces, and characteristic functions. For background, we refer interested readers to Durrett (2019) for measure theoretical probability, and Berlinet & Thomas-Agnan (2004) for reproducing kernel Hilbert spaces.

C.1 Proof of Proposition 4.1

Proposition. *Let U and V be stochastic processes on a common probability space with identical marginals but distinct path laws $\mathcal{D}_U$ and $\mathcal{D}_V$. For each $\lambda \in [0,1]$, define*

$$W_\lambda(t) = (1 - B_\lambda)U(t) + B_\lambda V(t), \qquad t \in (0, T],$$

where $B_\lambda \sim \text{Bernoulli}(\lambda)$ is independent of both processes and selects one process for the entire trajectory. The probability space may be enlarged to support this independent variable. The resulting path laws

$$\mathcal{D}_\lambda = (1 - \lambda)\mathcal{D}_U + \lambda\mathcal{D}_V, \qquad \lambda \in [0, 1],$$

form an uncountable family of distinct path laws sharing the same marginals.

Proof. Let U and V be the two processes, with path laws $\mathcal{D}_U \neq \mathcal{D}_V$. Enlarge their common probability space if necessary to support an independent Bernoulli variable. For each $\lambda \in [0,1]$, define

$$W_\lambda(t) = (1 - B_\lambda)U(t) + B_\lambda V(t), \qquad t \in (0, T],$$

where $B_\lambda \sim \text{Bernoulli}(\lambda)$ is independent of both processes and fixed across time. For every depth t and Borel set $A \subseteq \mathbb{R}^d$,

$$\begin{aligned}\mathbb{P}(W_\lambda(t) \in A) &= (1 - \lambda)\mathbb{P}(U(t) \in A) + \lambda\mathbb{P}(V(t) \in A) \\ &= \mathbb{P}(U(t) \in A),\end{aligned}$$

since U and V have identical marginals.

Because the same Bernoulli variable selects the entire trajectory, the path law of W_λ is

$$\mathcal{D}_\lambda = (1 - \lambda)\mathcal{D}_U + \lambda\mathcal{D}_V.$$

Since $\mathcal{D}_U \neq \mathcal{D}_V$, there is a measurable set of paths C such that $\mathcal{D}_U(C) \neq \mathcal{D}_V(C)$. For $\lambda \neq \lambda'$,

$$\mathcal{D}_\lambda(C) - \mathcal{D}_{\lambda'}(C) = (\lambda - \lambda')\big(\mathcal{D}_V(C) - \mathcal{D}_U(C)\big) \neq 0.$$

Thus distinct values of $\lambda \in [0,1]$ yield distinct path laws with the same marginals, giving an uncountable family. □

C.2 Proof of Theorem 4.3

Theorem (Full trajectory matching through consecutive groups)**.** *Let K_{Kol} be the kernel in Eq. 3 on $\mathcal{Z} = (\mathbb{R}^d)^{L+1}$, with fixed real valued, bounded, continuous, positive definite depth kernels k_ℓ, $\ell = 0, \ldots, L$, and fixed nonnegative weights satisfying $\sum_{0 \le a \le b \le L} w_{a,b} = 1$ and $w_{0,L} > 0$. Assume that each RKHS $\mathcal{H}_{k_\ell}$ is contained in and uniformly dense in $C_0(\mathbb{R}^d)$.*

Then K_{Kol} is bounded, continuous, and characteristic. For every $\theta \in \Theta$,

$$\mathcal{L}_{\text{Kol}}(\theta) = \text{MMD}^2_{K_{\text{Kol}}}(\mathcal{D}_\theta, \mathcal{D}_r).$$

In particular, $\mathcal{L}_{\text{Kol}}(\theta) = 0$ if and only if $\mathcal{D}_\theta = \mathcal{D}_r$.

Moreover, writing $I_{0:L} = (0, \ldots, L)$, for every sequence $(\theta_n) \subseteq \Theta$ and fixed reference law $\mathcal{D}_r$,

$$\mathcal{L}_{\text{Kol}}(\theta_n) \longrightarrow 0 \quad \Longleftrightarrow \quad \mathcal{L}_{I_{0:L}}(\theta_n) \longrightarrow 0 \quad \Longleftrightarrow \quad \mathcal{D}_{\theta_n} \Rightarrow \mathcal{D}_r,$$

where $\Rightarrow$ denotes weak convergence.

Proof. Throughout the proof, L, d, the depth kernels, the weights, and the reference law $\mathcal{D}_r$ are fixed. All measures are Borel probability measures unless otherwise specified. For any dimension p, $C_0(\mathbb{R}^p)$ denotes the space of continuous functions vanishing at infinity, equipped with the uniform norm $\|f\|_\infty = \sup_x |f(x)|$.

For an index tuple I, let π_I denote the corresponding coordinate projection. By definition,

$$\mu_\theta^I = (\pi_I)_\# \mathcal{D}_\theta, \qquad \mu_r^I = (\pi_I)_\# \mathcal{D}_r.$$

In particular, $\mu_\theta^{I_{0:L}} = \mathcal{D}_\theta$ and $\mu_r^{I_{0:L}} = \mathcal{D}_r$, so

$$\mathcal{L}_{I_{0:L}}(\theta) = \text{MMD}^2_{K_{I_{0:L}}}(\mathcal{D}_\theta, \mathcal{D}_r).$$

Kernel decomposition. For each interval $I_{a:b}$, lift its kernel to $\mathcal{Z}$ by

$$\overline{K}_{a:b}(z, z') = K_{I_{a:b}}(z_{a:b}, z'_{a:b}).$$

Each lifted kernel is positive definite, bounded, and continuous. Consequently, their nonnegative weighted sum

$$K_{\mathrm{Kol}} = \sum_{0 \le a \le b \le L} w_{a,b} \overline{K}_{a:b}$$

has the same properties.

Fix $\theta \in \Theta$ and write $\eta_\theta = \mathcal{D}_\theta - \mathcal{D}_r$. The squared MMD expression, linearity in the kernel, and the definition of a marginal give

$$\begin{aligned}
\mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\mathcal{D}_\theta, \mathcal{D}_r) &= \iint K_{\mathrm{Kol}}(z, z')\, d\eta_\theta(z)\, d\eta_\theta(z') \\
&= \sum_{a \le b} w_{a,b} \iint \overline{K}_{a:b}(z, z')\, d\eta_\theta(z)\, d\eta_\theta(z') \\
&= \sum_{a \le b} w_{a,b}\, \mathrm{MMD}^2_{K_{I_{a:b}}}\left(\mu_\theta^{I_{a:b}}, \mu_r^{I_{a:b}}\right) \\
&= \mathcal{L}_{\mathrm{Kol}}(\theta).
\end{aligned}$$

Every summand is nonnegative, so

$$w_{0,L} \mathcal{L}_{I_{0:L}}(\theta) \le \mathcal{L}_{\mathrm{Kol}}(\theta). \tag{8}$$

Since $w_{0,L} > 0$, this proves

$$\mathcal{L}_{\mathrm{Kol}}(\theta_n) \to 0 \quad \Longrightarrow \quad \mathcal{L}_{I_{0:L}}(\theta_n) \to 0.$$

Identification of the path law. Identify $\mathcal{Z}$ with $\mathbb{R}^D$, where $D = d(L+1)$, and write

$$K(z, z') = K_{I_{0:L}}(z, z') = \prod_{\ell=0}^{L} k_\ell(z_\ell, z'_\ell).$$

By assumption, each $\mathcal{H}_{k_\ell}$ is contained in and uniformly dense in $C_0(\mathbb{R}^d)$; equivalently, each k_ℓ is c_0-universal. Moreover, $k_\ell(x, \cdot) \in \mathcal{H}_{k_\ell} \subset C_0(\mathbb{R}^d)$ for every x, so each bounded kernel k_ℓ is a c_0-kernel. Since $\mathbb{R}^d$ is locally compact and Polish, Szabó & Sriperumbudur (2018, Theorem 5) implies that $\mathcal{H}_K$ is contained in and uniformly dense in $C_0(\mathbb{R}^D)$.

This density also makes K characteristic. Indeed, if two probability measures have the same mean embedding, their integrals agree for every function in $\mathcal{H}_K$. Uniform approximation then gives equality of their integrals for every function in $C_0(\mathbb{R}^D)$. Uniqueness in the Riesz representation theorem implies that the measures are equal.

The weighted MMD decomposition above holds for any pair of probability measures on $\mathcal{Z}$. Thus, zero MMD under K_{Kol} implies zero MMD under K, because $w_{0,L} > 0$. Since K is characteristic, so is K_{Kol}. In particular, $\mathcal{L}_{\mathrm{Kol}}(\theta) = 0$ if and only if $\mathcal{D}_\theta = \mathcal{D}_r$.

Equivalence with weak convergence. Now suppose $\mathcal{L}_{I_{0:L}}(\theta_n) \to 0$. Equivalently, $\mathrm{MMD}_K(\mathcal{D}_{\theta_n}, \mathcal{D}_r) \to 0$. For every $h \in \mathcal{H}_K$, the RKHS Cauchy–Schwarz inequality gives

$$\left| \int h\, d(\mathcal{D}_{\theta_n} - \mathcal{D}_r) \right| \le \|h\|_{\mathcal{H}_K}\, \mathrm{MMD}_K(\mathcal{D}_{\theta_n}, \mathcal{D}_r) \longrightarrow 0.$$

Let $f \in C_0(\mathbb{R}^D)$ and $\varepsilon > 0$. By uniform density, choose a fixed $h \in \mathcal{H}_K$ with $\|f - h\|_\infty < \varepsilon$. Since both measures have total mass one,

$$\left| \int f\, d(\mathcal{D}_{\theta_n} - \mathcal{D}_r) \right| \le 2\varepsilon + \|h\|_{\mathcal{H}_K}\, \mathrm{MMD}_K(\mathcal{D}_{\theta_n}, \mathcal{D}_r).$$

Taking the limit superior and then letting $\varepsilon \downarrow 0$ yields

$$\int f\, d\mathcal{D}_{\theta_n} \longrightarrow \int f\, d\mathcal{D}_r \qquad \text{for every } f \in C_0(\mathbb{R}^D). \tag{9}$$

To obtain weak convergence, we extend this conclusion to all bounded continuous functions. Since $\mathcal{D}_r$ is a probability measure on $\mathbb{R}^D$, for every $\varepsilon > 0$ there exists a continuous compactly supported function χ satisfying

$$0 \le \chi \le 1, \qquad \int \chi \, d\mathcal{D}_r > 1 - \varepsilon.$$

By Equation 9, $\int \chi \, d\mathcal{D}_{\theta_n} \to \int \chi \, d\mathcal{D}_r$, so for all sufficiently large n,

$$\int (1-\chi) \, d\mathcal{D}_{\theta_n} < 2\varepsilon, \qquad \int (1-\chi) \, d\mathcal{D}_r < \varepsilon.$$

For any bounded continuous f, the function $f\chi$ has compact support and belongs to $C_0(\mathbb{R}^D)$. Therefore,

$$\left| \int f \, d(\mathcal{D}_{\theta_n} - \mathcal{D}_r) \right| \le \left| \int f\chi \, d(\mathcal{D}_{\theta_n} - \mathcal{D}_r) \right| + \|f\|_\infty \left[\int (1-\chi) \, d\mathcal{D}_{\theta_n} + \int (1-\chi) \, d\mathcal{D}_r \right].$$

The first term tends to zero, while the remaining terms are eventually bounded by $3\varepsilon\|f\|_\infty$. Letting $\varepsilon \downarrow 0$ proves $\mathcal{D}_{\theta_n} \Rightarrow \mathcal{D}_r$.

Conversely, suppose $\mathcal{D}_{\theta_n} \Rightarrow \mathcal{D}_r$. For any interval $I = I_{a:b}$, the projection π_I is continuous. Thus, for every bounded continuous function g on the corresponding coordinate space,

$$\int g \, d\mu^I_{\theta_n} = \int g \circ \pi_I \, d\mathcal{D}_{\theta_n} \longrightarrow \int g \circ \pi_I \, d\mathcal{D}_r = \int g \, d\mu^I_r.$$

Hence $\mu^I_{\theta_n} \Rightarrow \mu^I_r$, and consequently

$$\mu^I_{\theta_n} \otimes \mu^I_{\theta_n} \Rightarrow \mu^I_r \otimes \mu^I_r, \qquad \mu^I_{\theta_n} \otimes \mu^I_r \Rightarrow \mu^I_r \otimes \mu^I_r.$$

For completeness, let φ_n and φ be the characteristic functions of $\mu^I_{\theta_n}$ and μ^I_r, respectively. The characteristic functions of the two product measures on the left are $\varphi_n(s)\varphi_n(t)$ and $\varphi_n(s)\varphi(t)$. Both converge pointwise to $\varphi(s)\varphi(t)$, the characteristic function of $\mu^I_r \otimes \mu^I_r$. The assertions follow from the continuity theorem for characteristic functions.

Since K_I is bounded and continuous,

$$\begin{aligned} \mathcal{L}_I(\theta_n) &= \mathrm{MMD}^2_{K_I}(\mu^I_{\theta_n}, \mu^I_r) \\ &= \int K_I \, d(\mu^I_{\theta_n} \otimes \mu^I_{\theta_n}) + \int K_I \, d(\mu^I_r \otimes \mu^I_r) \\ &\quad - 2 \int K_I \, d(\mu^I_{\theta_n} \otimes \mu^I_r) \longrightarrow 0. \end{aligned}$$

There are only finitely many intervals, and their weights are fixed. Therefore,

$$\mathcal{L}_{\mathrm{Kol}}(\theta_n) = \sum_{0 \le a \le b \le L} w_{a,b} \mathcal{L}_{I_{a:b}}(\theta_n) \longrightarrow 0.$$

We have proved

$$\mathcal{L}_{\mathrm{Kol}}(\theta_n) \to 0 \quad \Longrightarrow \quad \mathcal{L}_{I_{0:L}}(\theta_n) \to 0 \quad \Longrightarrow \quad \mathcal{D}_{\theta_n} \Rightarrow \mathcal{D}_r \quad \Longrightarrow \quad \mathcal{L}_{\mathrm{Kol}}(\theta_n) \to 0,$$

which establishes all three equivalences and completes the proof. □

C.3 Proof of Theorem 4.5

Theorem (Functional improvement over marginal matching)**.** *Let $X^{(0)} \sim \mu^{(0)}$ be the common input to the steered and reference trajectories. Denote their path laws by*

$$\mathcal{D}_\theta = \mathrm{Law}(\widetilde{\mathcal{X}}_\theta), \qquad \mathcal{D}_r = \mathrm{Law}(\mathcal{X}_r).$$

For every fixed θ, assume both trajectories are deterministic conditional on $X^{(0)}$, so their randomness comes only from sampling the input. The steered trajectory is produced by applying the

interventions during the network's forward pass. Write h_θ and f for the output functions, so that $\widetilde{X}_\theta^{(L)} = h_\theta(X^{(0)})$ and $X_r^{(L)} = f(X^{(0)})$. Suppose these laws, for all $\theta \in \Theta$, are supported on a common compact set $\mathcal{Z} \subseteq (\mathbb{R}^d)^{L+1}$, and f is continuous on its input projection. Define the functional error by

$$R(\theta) = \mathbb{E}_{X^{(0)}\sim\mu^{(0)}}\left[\|h_\theta(X^{(0)}) - f(X^{(0)})\|^2\right].$$

Consider the population objectives

$$\mathcal{L}_{\mathrm{marg}}(\theta) = \sum_{\ell=1}^{L} \mathrm{MMD}^2_{k_\ell}\big(\mu_\theta^{(\ell)}, \mu_r^{(\ell)}\big),$$
$$\mathcal{L}_{\mathrm{Kol}}(\theta) = \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\mathcal{D}_\theta, \mathcal{D}_r),$$

with the kernels and weights defined above, so that K_{Kol} is continuous and characteristic.

Suppose the steering family contains a parameter θ^{marg} satisfying

$$\mu_{\theta^{\mathrm{marg}}}^{(\ell)} = \mu_r^{(\ell)} \quad \textit{for every } \ell \in \{1,\dots,L\}, \qquad R(\theta^{\mathrm{marg}}) \geq \Delta > 0.$$

Then θ^{marg} is a global minimizer of $\mathcal{L}_{\mathrm{marg}}$.

For $z = (z_0,\dots,z_L) \in \mathcal{Z}$, set $r(z) = \|z_L - f(z_0)\|^2$. Define

$$\tau_\Delta := \inf_{\substack{\nu\in\mathcal{P}(\mathcal{Z}) \\ \int_{\mathcal{Z}} r\, d\nu \geq \Delta}} \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\nu, \mathcal{D}_r),$$

where $\mathcal{P}(\mathcal{Z})$ denotes the Borel probability measures on $\mathcal{Z}$. Thus τ_Δ is the infimum of the discrepancy from the reference law among trajectory laws with expected error at least Δ. Then $\tau_\Delta > 0$. If, for $a, \varepsilon \geq 0$,

$$\inf_{\theta\in\Theta} \mathcal{L}_{\mathrm{Kol}}(\theta) \leq a, \qquad \mathcal{L}_{\mathrm{Kol}}(\hat{\theta}) \leq \inf_{\theta\in\Theta} \mathcal{L}_{\mathrm{Kol}}(\theta) + \varepsilon,$$

and $a + \varepsilon < \tau_\Delta$, then

$$R(\hat{\theta}) < \Delta \leq R(\theta^{\mathrm{marg}}).$$

The same threshold applies to every marginal matching solution whose functional error is at least Δ.

The threshold τ_Δ connects the Kolmogorov objective to functional error: for every $\theta \in \Theta$,

$$\mathcal{L}_{\mathrm{Kol}}(\theta) < \tau_\Delta \quad \Longrightarrow \quad R(\theta) < \Delta.$$

The proof below constructs this threshold and establishes its positivity.

Proof. All expectations below are over $X^{(0)} \sim \mu^{(0)}$. For each fixed θ, the steered and reference trajectories contain no additional randomness once the input is specified.

Since the baseline matches every layer marginal,

$$\mu_{\theta^{\mathrm{marg}}}^{(\ell)} = \mu_r^{(\ell)} \qquad \text{for every } \ell,$$

each term in its marginal matching objective is zero. Thus

$$\mathcal{L}_{\mathrm{marg}}(\theta^{\mathrm{marg}}) = 0.$$

Every squared MMD is nonnegative, so zero is the smallest possible objective value. Hence θ^{marg} is a global minimizer of the marginal matching objective, despite having functional error $R(\theta^{\mathrm{marg}}) \geq \Delta > 0$.

Recall the trajectory error function from the theorem. For $z = (z_0,\dots,z_L) \in \mathcal{Z}$,

$$r(z) = \|z_L - f(z_0)\|^2.$$

Because f is continuous, r is continuous. Since $\mathcal{Z}$ is compact, r is also bounded. The laws $\mathcal{D}_\theta = \mathrm{Law}(\widetilde{\mathcal{X}}_\theta)$ and $\mathcal{D}_r = \mathrm{Law}(\mathcal{X}_r)$ assign probabilities to entire random trajectories. Both trajectories preserve the same input, so $\widetilde{X}_\theta^{(0)} = X_r^{(0)} = X^{(0)}$. Consequently,

$$\begin{aligned}\int_{\mathcal{Z}} r\, d\mathcal{D}_\theta &= \mathbb{E}\big[r(\widetilde{\mathcal{X}}_\theta)\big] \\ &= \mathbb{E}\big\|\widetilde{X}_\theta^{(L)} - f(\widetilde{X}_\theta^{(0)})\big\|^2 \\ &= \mathbb{E}\|h_\theta(X^{(0)}) - f(X^{(0)})\|^2 = R(\theta).\end{aligned}$$

For the reference trajectory, $X_r^{(L)} = f(X^{(0)})$, and hence

$$\begin{aligned}\int_{\mathcal{Z}} r\, d\mathcal{D}_r &= \mathbb{E}\big[r(\mathcal{X}_r)\big] \\ &= \mathbb{E}\big\|X_r^{(L)} - f(X_r^{(0)})\big\|^2 \\ &= \mathbb{E}\|f(X^{(0)}) - f(X^{(0)})\|^2 = 0.\end{aligned}$$

Consider all trajectory distributions whose functional error is at least Δ:

$$\mathcal{C}_\Delta = \left\{\nu \in \mathcal{P}(\mathcal{Z}) : \int_{\mathcal{Z}} r\, d\nu \geq \Delta\right\}.$$

This set is nonempty because it contains $\mathcal{D}_{\theta^{\mathrm{marg}}}$. The threshold defined in the theorem can therefore be written as

$$\tau_\Delta = \inf_{\nu \in \mathcal{C}_\Delta} \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\nu, \mathcal{D}_r).$$

We first prove that $\tau_\Delta > 0$. Suppose instead that $\tau_\Delta = 0$. By the definition of the infimum, for every $n \geq 1$ there exists $\nu_n \in \mathcal{C}_\Delta$ such that

$$\mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\nu_n, \mathcal{D}_r) < \frac{1}{n}.$$

Consequently,

$$\mathrm{MMD}_{K_{\mathrm{Kol}}}(\nu_n, \mathcal{D}_r) \to 0.$$

Since $\mathcal{Z}$ is a compact metric space, it is complete and separable, hence Polish. The sequence (ν_n) is tight: for every $\varepsilon > 0$, the same compact set $K_\varepsilon = \mathcal{Z}$ satisfies $\nu_n(K_\varepsilon) = 1 \geq 1 - \varepsilon$ for all n.

Prokhorov's theorem states that every tight sequence of Borel probability measures on a Polish space has a weakly convergent subsequence. It therefore gives a subsequence ν_{n_j} converging weakly to some $\nu_* \in \mathcal{P}(\mathcal{Z})$.

The kernel K_{Kol} is bounded and continuous, so weak convergence of the corresponding product measures in the squared MMD expression gives

$$\mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\nu_*, \mathcal{D}_r) = \lim_{j\to\infty} \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\nu_{n_j}, \mathcal{D}_r) = 0.$$

Characteristicness implies $\nu_* = \mathcal{D}_r$. Since r is bounded and continuous, weak convergence then gives

$$\int_{\mathcal{Z}} r\, d\nu_{n_j} \longrightarrow \int_{\mathcal{Z}} r\, d\mathcal{D}_r = 0.$$

But each $\nu_{n_j} \in \mathcal{C}_\Delta$ satisfies $\int r\, d\nu_{n_j} \geq \Delta > 0$, a contradiction. Thus $\tau_\Delta > 0$.

Now let $\hat{\theta}$ be the learned parameters. The assumed approximation and optimization bounds yield

$$\begin{aligned}\mathcal{L}_{\mathrm{Kol}}(\hat{\theta}) &\leq \inf_{\theta\in\Theta} \mathcal{L}_{\mathrm{Kol}}(\theta) + \varepsilon \\ &\leq a + \varepsilon \\ &< \tau_\Delta.\end{aligned}$$

If $R(\hat{\theta}) \geq \Delta$, then $\mathcal{D}_{\hat{\theta}} \in \mathcal{C}_\Delta$. By the definition of τ_Δ, this would imply

$$\mathcal{L}_{\mathrm{Kol}}(\hat{\theta}) = \mathrm{MMD}^2_{K_{\mathrm{Kol}}}(\mathcal{D}_{\hat{\theta}}, \mathcal{D}_r) \geq \tau_\Delta,$$

contradicting the preceding inequality. Therefore,

$$R(\hat{\theta}) < \Delta \leq R(\theta^{\text{marg}}).$$

Finally, τ_Δ depends only on the reference distribution, the kernel, the compact domain, and Δ. It does not depend on the particular baseline, so the same comparison holds for every marginal matching solution with functional error at least Δ. □

C.4 Proof of Theorem 4.7

Theorem (Exact recovery of the difference in means vector)**.** *Let $X_s, X_r \in \mathbb{R}^d$ be source and reference activations at a fixed layer, with finite first moments $m_s = \mathbb{E}[X_s]$ and $m_r = \mathbb{E}[X_r]$. Suppose their centered distributions are symmetric positive definite (SPD), meaning that their characteristic functions are real and strictly positive at every frequency (Law et al., 2017). For any real valued, bounded, continuous, translation invariant characteristic kernel k, the unique optimal shift for the single layer marginal MMD objective is*

$$\arg\min_{b \in \mathbb{R}^d} \text{MMD}_k^2\big(\text{Law}(X_s + b), \text{Law}(X_r)\big) = \{m_r - m_s\}.$$

Proof. For a random vector Z, write $\phi_Z(\omega) = \mathbb{E}[e^{i\omega^\top Z}]$ for its characteristic function. Set $b_0 = m_r - m_s$, $\delta = b - b_0$, and

$$\mathcal{L}(b) = \text{MMD}_k^2\big(\text{Law}(X_s + b), \text{Law}(X_r)\big).$$

Since k is translation invariant, write $k(x, y) = \psi(x - y)$, where ψ is continuous and positive definite. Bochner's theorem (Sriperumbudur et al., 2010, Theorem 3) gives a finite nonnegative Borel measure Λ such that

$$k(x, y) = \int_{\mathbb{R}^d} e^{i\omega^\top (x-y)}\, d\Lambda(\omega).$$

In particular, $\Lambda(A) \geq 0$ for every Borel set A, and $\Lambda(\mathbb{R}^d) = \psi(0) < \infty$.

Since k is also characteristic, Theorem 9 of Sriperumbudur et al. (2010) implies $\text{supp}(\Lambda) = \mathbb{R}^d$. Thus every nonempty open ball has strictly positive Λ-measure.

Corollary 4 of Sriperumbudur et al. (2010) now gives the spectral representation of MMD:

$$\mathcal{L}(b) = \int_{\mathbb{R}^d} |\phi_{X_s+b}(\omega) - \phi_{X_r}(\omega)|^2 \, d\Lambda(\omega).$$

All integrals below are finite because characteristic functions have absolute value at most one and Λ is finite. Define the centered characteristic functions by $\phi_s^\circ = \phi_{X_s - m_s}$ and $\phi_r^\circ = \phi_{X_r - m_r}$. Separating the deterministic means from the centered random vectors yields

$$\phi_{X_s+b}(\omega) = \mathbb{E}\Big[e^{i\omega^\top ((X_s - m_s) + (m_s + b))}\Big] = e^{i\omega^\top (m_s + b)} \phi_s^\circ(\omega),$$
$$\phi_{X_r}(\omega) = \mathbb{E}\Big[e^{i\omega^\top ((X_r - m_r) + m_r)}\Big] = e^{i\omega^\top m_r} \phi_r^\circ(\omega).$$

Since $m_s + b = m_r + \delta$, factoring out $e^{i\omega^\top m_r}$, whose absolute value is one, gives

$$\mathcal{L}(b) = \int_{\mathbb{R}^d} \Big| e^{i\omega^\top \delta} \phi_s^\circ(\omega) - \phi_r^\circ(\omega) \Big|^2 \, d\Lambda(\omega).$$

By the SPD assumption, $\phi_s^\circ(\omega)$ and $\phi_r^\circ(\omega)$ are real and strictly positive for every $\omega \in \mathbb{R}^d$. Expanding the squared absolute value therefore gives

$$\Big| e^{i\omega^\top \delta} \phi_s^\circ(\omega) - \phi_r^\circ(\omega) \Big|^2$$
$$= \big(\phi_s^\circ(\omega)\big)^2 + \big(\phi_r^\circ(\omega)\big)^2 - 2\phi_s^\circ(\omega)\phi_r^\circ(\omega) \cos(\omega^\top \delta).$$

At $b = b_0$, we have $\delta = 0$ and the cosine equals one. Subtracting the corresponding expression for $\mathcal{L}(b_0)$ thus yields

$$\mathcal{L}(b) - \mathcal{L}(b_0) = 2 \int_{\mathbb{R}^d} \phi_s^\circ(\omega)\phi_r^\circ(\omega)\big[1 - \cos(\omega^\top \delta)\big]\, d\Lambda(\omega) \geq 0.$$

The inequality follows because the characteristic functions are positive, $1 - \cos(\omega^\top \delta) \geq 0$, and Λ is a nonnegative measure. Hence b_0 is a global minimizer.

To prove uniqueness, suppose $\delta \neq 0$ and define

$$F_\delta(\omega) = 2\phi_s^\circ(\omega)\phi_r^\circ(\omega)\big[1 - \cos(\omega^\top \delta)\big].$$

Choose $\omega_* = \pi\delta/\|\delta\|^2$. Then $\omega_*^\top \delta = \pi$, so

$$F_\delta(\omega_*) = 4\phi_s^\circ(\omega_*)\phi_r^\circ(\omega_*) > 0.$$

Characteristic functions are continuous, so F_δ is continuous. There consequently exist an open ball U around ω_* and a constant $c > 0$ such that $F_\delta(\omega) \geq c$ for every $\omega \in U$. Full spectral support gives $\Lambda(U) > 0$. Since F_δ is nonnegative everywhere,

$$\mathcal{L}(b) - \mathcal{L}(b_0) = \int_{\mathbb{R}^d} F_\delta(\omega)\, d\Lambda(\omega) \geq \int_U F_\delta(\omega)\, d\Lambda(\omega) \geq c\,\Lambda(U) > 0.$$

Thus every $b \neq b_0$ has strictly larger loss, proving that $b_0 = m_r - m_s$ is the unique global minimizer of the single layer marginal MMD objective. □

C.5 Cached fitting and functional guarantees

We distinguish the law of transformed cached activations from the law produced by applying the same maps during a forward pass. This distinction does not affect distributional identification: under the stated kernel conditions, MMD identifies the law it compares. It does affect the interpretation of that law and its relationship to model outputs.

Why the input and the actual trajectory matter. Two examples show why the assumptions of Theorem 4.5 cannot simply be removed. First, let U be uniform on $\{-1, 1\}$. Reference internal states (U, U) and steered internal states $(-U, -U)$ have identical joint laws, although their terminal values have expected squared difference $\mathbb{E}|U - (-U)|^2 = 4$ on the same input. Including the input distinguishes the laws of (U, U, U) and $(U, -U, -U)$. Thus, even exact matching of actual internal trajectories does not generally determine the output associated with each input.

Second, consider two identity layers with unsteered trajectory (U, U, U) and maps $S_1(x) = x + a$ and $S_2(x) = x - a$, where $a \neq 0$. Applying the maps to cached activations gives $(U, U + a, U - a)$, whereas applying them during the forward pass gives $(U, U + a, U)$. Take $(U, U + a, U - a)$ as the reference trajectory, which is generated by the reference layers $x \mapsto x + a$ and $x \mapsto x - 2a$. The cached loss is zero, even with the input included, but the actual terminal output has squared error a^2. This example separates the cached approximation from the omission of the input.

From cached discrepancy to actual discrepancy. Let $\bar{Z}_\theta$ be the tuple of transformed cached activations and Z_θ the tuple recorded at the same sites during an actual intervened forward pass, coupled through the same source input. Denote their laws by $\bar{\mathcal{D}}_\theta$ and $\mathcal{D}_\theta^{\mathrm{act}}$, and let $\mathcal{D}_r$ be the reference law on this tuple space. For a fixed bounded positive definite kernel K with RKHS $\mathcal{H}_K$, write $\Phi_K(z) = K(z, \cdot)$ and $m_K(\nu) = \mathbb{E}_{Z\sim\nu}[\Phi_K(Z)]$. Define

$$\mathcal{L}_{\mathrm{cache}}(\theta) = \mathrm{MMD}_K^2(\bar{\mathcal{D}}_\theta, \mathcal{D}_r), \qquad \mathcal{L}_{\mathrm{act}}(\theta) = \mathrm{MMD}_K^2(\mathcal{D}_\theta^{\mathrm{act}}, \mathcal{D}_r).$$

These are population discrepancies on the same recorded coordinates; the tuple need not include the input.

Proposition C.1 (Error from cached approximation)**.** *Suppose* $\|\Phi_K(z) - \Phi_K(z')\|_{\mathcal{H}_K} \leq c_K\|z - z'\|_2$ *and* $\delta_\theta = (\mathbb{E}\|Z_\theta - \bar{Z}_\theta\|_2^2)^{1/2} < \infty$. *Then*

$$\left|\sqrt{\mathcal{L}_{\mathrm{act}}(\theta)} - \sqrt{\mathcal{L}_{\mathrm{cache}}(\theta)}\right| \leq c_K\delta_\theta. \tag{10}$$

Let $\widehat{\bar{\mathcal{D}}}_\theta$ and $\widehat{\mathcal{D}}_r$ be the empirical laws used for fitting, and let $\widehat{\mathcal{L}}_{\mathrm{cache}}(\theta) = \mathrm{MMD}_K^2(\widehat{\bar{\mathcal{D}}}_\theta, \widehat{\mathcal{D}}_r)$. Define their total embedding estimation error by

$$g_\theta = \|m_K(\bar{\mathcal{D}}_\theta) - m_K(\widehat{\bar{\mathcal{D}}}_\theta)\|_{\mathcal{H}_K} + \|m_K(\mathcal{D}_r) - m_K(\widehat{\mathcal{D}}_r)\|_{\mathcal{H}_K}.$$

Then

$$\sqrt{\mathcal{L}_{\mathrm{act}}(\theta)} \le \sqrt{\widehat{\mathcal{L}}_{\mathrm{cache}}(\theta)} + c_K\delta_\theta + g_\theta. \tag{11}$$

Proof. Using the coupling through the source input, Jensen's inequality and Cauchy–Schwarz give

$$\begin{aligned}\mathrm{MMD}_K(\mathcal{D}_\theta^{\mathrm{act}}, \bar{\mathcal{D}}_\theta) &= \|\mathbb{E}[\Phi_K(Z_\theta) - \Phi_K(\bar{Z}_\theta)]\|_{\mathcal{H}_K} \\ &\le c_K\mathbb{E}\|Z_\theta - \bar{Z}_\theta\|_2 \le c_K\delta_\theta.\end{aligned}$$

The reverse triangle inequality for the embedding norm proves Equation 10. Applying the triangle inequality to the two empirical embedding errors gives Equation 11. □

For the Gaussian and rational quadratic interval kernels in Appendix B, this feature map condition holds with finite positive scales. Index the recorded sites here by $1, \dots, J$. For $K = \sum_I w_I k_I$, with nonnegative weights summing to one, site scales s_j, interval bandwidths τ_I, and interval length $|I|$, one may take

$$c_K^2 = \max_{1\le j\le J}\left[\frac{1}{s_j^2}\sum_{I\ni j}\frac{w_I}{|I|\tau_I^2}\right]. \tag{12}$$

Indeed, both interval kernels have diagonal value one and satisfy $2(1 - k_I(z_I, z_I')) \le \sum_{j\in I}\|z_j - z_j'\|_2^2/(|I|\tau_I^2 s_j^2)$. For the Gaussian kernel this follows from $1 - e^{-t} \le t$; for the rational quadratic kernel it follows from $(1+t)^{-\alpha} \ge 1 - \alpha t$ for $t \ge 0$ and $\alpha > 0$. Summing over intervals and using $\|\Phi_K(z) - \Phi_K(z')\|_{\mathcal{H}_K}^2 = 2(K(z,z) - K(z,z'))$ gives Equation 12.

A sufficient stability condition. The approximation error can itself be bounded for Lipschitz transitions and maps. Fix θ and suppress it in the notation. For complete network states, write

$$\begin{aligned}X_\ell &= F_\ell(X_{\ell-1}), \qquad \bar{Y}_\ell = S_\ell(X_\ell), \\ Y_\ell &= S_\ell(F_\ell(Y_{\ell-1})), \quad Y_0 = \bar{Y}_0 = X_0.\end{aligned}$$

Here X_ℓ is the unsteered state, $\bar{Y}_\ell$ the transformed cached state, and Y_ℓ the actual intervened state. Suppose F_ℓ and S_ℓ have Lipschitz constants A_ℓ and B_ℓ, respectively. Let $b_j = (\mathbb{E}\|S_j(X_j) - X_j\|_2^2)^{1/2} < \infty$ be the size of the intervention on the cached state. Then, with an empty sum equal to zero,

$$\begin{aligned}e_\ell := (\mathbb{E}\|Y_\ell - \bar{Y}_\ell\|_2^2)^{1/2} &\le \sum_{j=1}^{\ell-1} b_j \prod_{k=j+1}^{\ell} A_k B_k, \\ \delta_\theta &\le \left(\sum_{\ell=1}^{L} e_\ell^2\right)^{1/2}.\end{aligned} \tag{13}$$

The second inequality applies when recorded activations are coordinate projections of these states, which cannot increase Euclidean distance. To prove the first, set $e_0 = b_0 = 0$. Lipschitz continuity and the triangle inequality in L^2 give $e_\ell \le A_\ell B_\ell(e_{\ell-1} + b_{\ell-1})$; expanding this recurrence yields the bound. A complete state must contain all quantities needed for the next transition, including residual branches. The recurrence does not assume that the recorded activation subset alone determines later states. The products of Lipschitz constants can grow with depth, so this bound need not be small. The hard sign mask in AAS can be discontinuous, so the propagation bound requires additional assumptions on its activation domain; otherwise, using Proposition C.1 requires separate control of δ_θ.

Scope of the functional conclusion. If the matched tuples include the shared input and satisfy all assumptions of Theorem 4.5, Equation 11 gives a sufficient condition for its conclusion:

$$\left(\sqrt{\widehat{\mathcal{L}}_{\text{cache}}(\theta)} + c_K\delta_\theta + g_\theta\right)^2 < \tau,$$

where τ is the theorem's population threshold. This condition separates empirical matching, cached approximation, and embedding estimation error. For a learned map and a kernel selected from data, the estimation error g_θ also requires control after selection; the inequality alone supplies no such guarantee. Our experiments omit the input and may use different source and reference prompts, and we do not establish small δ_θ or g_θ for those fits. Their results therefore demonstrate the practical utility of cached activation matching, rather than validate the functional theorem's assumptions. Extending the guarantee to unpaired prompts requires additional structure relating the inputs or their desired outputs, which internal distribution matching alone does not supply.

D JAILBREAK EXPERIMENT

Activation steering methods can be broadly divided into two phases: extraction and inference. During extraction, the method determines the steering map parameters from examples of the source and reference behaviors. During inference, the resulting map is applied as the model processes new requests, without knowing their behavior labels in advance. Below, we describe both phases for each steering method. Kolmogorov steering differs from its baseline in how the map parameters are obtained, while retaining the same inference time intervention procedure.

D.1 EXPERIMENTAL SETUP

This task aims to bypass a model's refusal mechanism and steer it to answer harmful requests. The source and reference distributions correspond to activation trajectories induced by harmful and harmless requests, respectively.

Models. We evaluate three models: Qwen3.5-27B (Qwen Team, 2026), GPT-OSS-20B (Agarwal et al., 2025), and Gemma-4-31B-IT (Team, 2026).

Metrics. We measure jailbreak effectiveness using attack success rates (ASR) judged by `cais/HarmBench-Llama-2-13b-cls`[2] (Mazeika et al., 2024) and retention of general model capabilities using TinyBench scores (Maia Polo et al., 2024). We use the 159 standard examples from the official HarmBench text test split for validation and parameter selection, and the 104 held-out AdvBench prompts for final testing. Table 1 reports the frozen test results for AAS, DirAbl, and their Kolmogorov variants. Each steered configuration has 104 test judgments with no invalid classifier labels. The separate CAA analysis retains its original protocol in Appendix F.

Baselines. We consider two steering methods representing rotational and projection maps, respectively:

- **Adaptive Angular Steering (AAS)** (Vu & Nguyen, 2025). AAS selectively rotates activations within a two dimensional steering plane while preserving their norms and components orthogonal to that plane. Let $\hat{d}_{\text{feat}}$ denote the unit feature direction used for masking, and let $B = [b_1\ b_2]$ contain an orthonormal basis of the steering plane P. The steering map is

$$S_{\theta,\alpha}(x) = x + \text{mask}(x)\left(\|\,\text{proj}_P(x)\|_2 B R_\alpha \begin{bmatrix}1\\0\end{bmatrix} - \text{proj}_P(x)\right),$$

  where $\text{proj}_P(x) = BB^\top x$, $\text{mask}(x) = \max(0, \text{sign}(\hat{d}_{\text{feat}}^\top x))$, and R_α is the two dimensional rotation matrix for the target angle α. The extracted parameters are $\theta = (\hat{d}_{\text{feat}}, b_1, b_2)$. During extraction, the original method computes difference in means (DIM) vectors between harmful and harmless activations at multiple extraction points. It selects

[2]HarmBench classifier checkpoint: https://huggingface.co/cais/HarmBench-Llama-2-13b-cls.

the candidate with the highest average cosine similarity to the others and normalizes it to obtain $\hat{d}_{\text{feat}}$. The steering plane is constructed from this direction and the first principal component of the candidate vectors, with Gram–Schmidt orthogonalization yielding the basis B. In the original construction, $b_1 = \hat{d}_{\text{feat}}$. The target angle α is an inference time hyperparameter selected separately. AAS therefore represents an adaptive rotational map: the feature direction determines which activations are modified, while the plane basis and target angle determine the rotation.

- **Directional Ablation (DirAbl)** (Arditi et al., 2024). DirAbl removes the component of an activation along an extracted refusal direction:

$$S_\theta(x) = (I - uu^\top)x, \qquad \theta = u, \quad \|u\|_2 = 1.$$

During extraction, the method computes DIM vectors d_i at candidate layer–token locations. These vectors are evaluated to select a direction whose removal effectively suppresses refusal. In our evaluation, we search over candidate extraction locations on the HarmBench validation set to identify the strongest performing heuristic baseline. The selected vector d_{i^*} is normalized as $u = d_{i^*}/\|d_{i^*}\|_2$. DirAbl therefore represents an orthogonal projection onto the subspace perpendicular to the refusal direction.

Proposed methods. For each steering baseline, Kolmogorov Steering replaces heuristic parameter selection with parameters learned by minimizing the proposed MMD loss in Eq. 2.

Steering details. Following Vu & Nguyen (2025), we construct the source and reference activation distributions using 416 harmful prompts from AdvBench (Zou et al., 2023) and 512 harmless Alpaca instructions (Taori et al., 2023) with no auxiliary input, respectively. AdvBench is partitioned once into an 80/20 training/test split using random seed 42. The remaining 104 harmful prompts are excluded from fitting and configuration selection in this experiment and serve as the frozen test set. We use the 159 standard behaviors from the official HarmBench text test split (Mazeika et al., 2024) as our validation set. Steering maps and inference time settings are frozen after validation selection, then evaluated on AdvBench and TinyBench without further fitting or parameter selection.

For AAS and DirAbl, we extract normalized activations from an ordered sequence of layer normalization sites across nonfinal decoder blocks. For Qwen3.5-27B and GPT-OSS-20B, we use `input_layernorm` followed by `post_attention_layernorm` in each block, covering layers 0–62 and 0–22, respectively. For Gemma-4-31B-IT, we use the chronological pair `pre_feedforward_layernorm`$[L]$ and `input_layernorm`$[L+1]$, with $L = 0, \ldots, 58$. At inference time, AAS and DirAbl intervene across multiple layers at these layer normalization sites; the selected Kolmogorov DirAbl map for Gemma uses layers 10–48. Extraction and intervention use the same activation sites, and the map parameters remain fixed during inference.

For Qwen3.5-27B and Gemma-4-31B-IT, we capture activations at the final token of the fully formatted generation prompt, immediately before response generation, with thinking disabled. For GPT-OSS-20B, we use the Harmony template with reasoning effort set to `low` and capture activations at the first `<|message|>` token, immediately before the model begins generating analysis content.

Map parameter selection. For the heuristic baselines, we search over the target angle for AAS and DIM directions from all eligible steering locations for DirAbl. We choose the configuration with the highest ASR, judged by the HarmBench classifier, on the 159 HarmBench validation prompts. These searches aim to find the best-performing setting within each evaluated heuristic parameterization. TinyBench is used only to measure capability retention, not to select configurations. CAA is excluded from selection and testing in this experiment.

For Kolmogorov steering, we train the map parameters using only the 416 harmful and 512 harmless training trajectories. We use Adam (Kingma & Ba, 2014) for 3,000 steps, starting with a learning rate of 0.01 and reducing it to zero with a cosine schedule (Loshchilov & Hutter, 2016). Each step uses 32 harmful trajectories and all 512 harmless reference trajectories. We use random seed 0, with no weight decay or early stopping. The loss is the biased empirical squared MMD, including diagonal pairs, with a Gaussian interval kernel over all contiguous trajectory intervals. All intervals receive equal weight, and the interval bandwidth parameter is $\tau = 1$. Before training, we fix the

kernel scale at each activation point using the median heuristic on at most 4,096 reference pairs. Pairwise kernel sums are computed in blocks of 256.

For Kolmogorov AAS, we initialize the steering plane basis B and feature direction $\hat{d}_{\text{feat}}$ from the heuristic AAS solution. We train B while keeping $\hat{d}_{\text{feat}}$ fixed so that the optimizer cannot avoid applying steering by changing the masking direction. For Kolmogorov DirAbl, we initialize the direction u randomly.

After fitting, the target angle for Kolmogorov AAS is selected on the same 159 HarmBench validation prompts. Kolmogorov DirAbl uses the established fitted maps at full projection strength, without an additional strength sweep. Validation scores do not enter the MMD objective or gradient updates. The frozen AAS and DirAbl configurations are then evaluated on the 104 AdvBench test prompts only through response generation and ASR classification.

D.2 Map selection results

The following sweeps use the 159 standard examples from the official HarmBench text test split as validation prompts, with ASR judged by `cais/HarmBench-Llama-2-13b-cls`. Their maxima describe validation performance over the evaluated grids, not final test performance. Table 1 reports AAS and DirAbl results on the 104 AdvBench test prompts after all configurations are frozen. The AdvBench test ASRs are not used for configuration selection. CAA is excluded from this experiment.

DirAbl direction selection. We compare candidate refusal directions extracted at different layer–site locations. The selected directions come from `input_layernorm` at layer 20 for Qwen3.5-27B (validation ASR 31.45%), `input_layernorm` at layer 17 for GPT-OSS-20B (77.36%), and `pre_feedforward_layernorm` at layer 35 for Gemma-4-31B-IT (83.02%). These frozen original DirAbl configurations achieve test ASRs of 8.65%, 78.85%, and 83.65%, respectively, compared with 73.08%, 93.27%, and 90.38% for Kolmogorov DirAbl. These locations identify where the direction is extracted, not a restriction of inference time ablation to a single layer. Kolmogorov DirAbl uses the established joint-site fitted maps at full strength, including the last-token map for GPT-OSS-20B and the randomly initialized layer 10–48 map for Gemma-4-31B-IT.

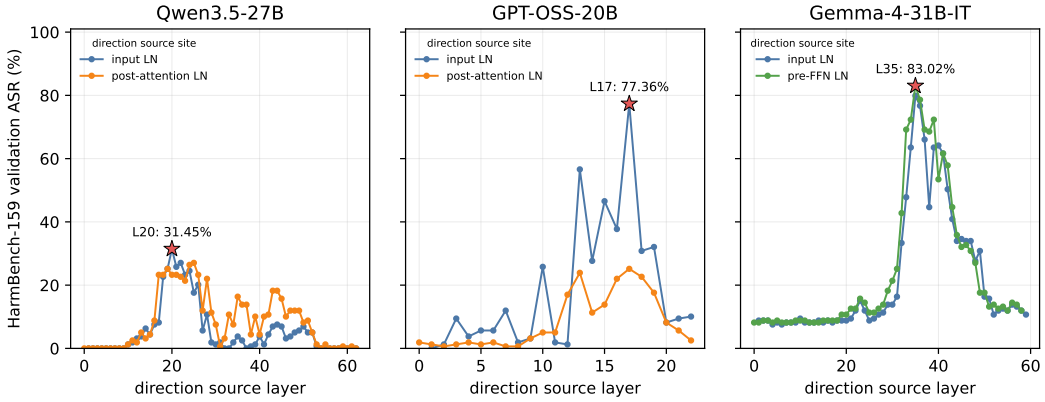


Figure A2: DirAbl direction selection on the HarmBench-159 validation set. Colors indicate extraction sites; stars mark the selected directions.

AAS angle selection. For AAS, we vary the inference time target angle while keeping the extracted or fitted steering plane fixed. On Qwen3.5-27B, the selected angles are 190° for AAS and 160° for Kolmogorov AAS, with validation ASRs of 88.05% and 91.82%, respectively. On GPT-OSS-20B, the selected angles are 270° and 110°, with validation ASRs of 92.45% and 90.57%. On Gemma-4-31B-IT, the selected angles are 210° and 190°, with validation ASRs of 83.02% and 87.42%. These results show that the target angle should be selected for the fitted plane rather than reused unchanged from the heuristic baseline. On the separate AdvBench test set, Kolmogorov AAS

improves ASR by 4.81 percentage points on GPT-OSS-20B and 17.31 points on Gemma-4-31B-IT, while scoring 0.96 points below original AAS on Qwen3.5-27B.

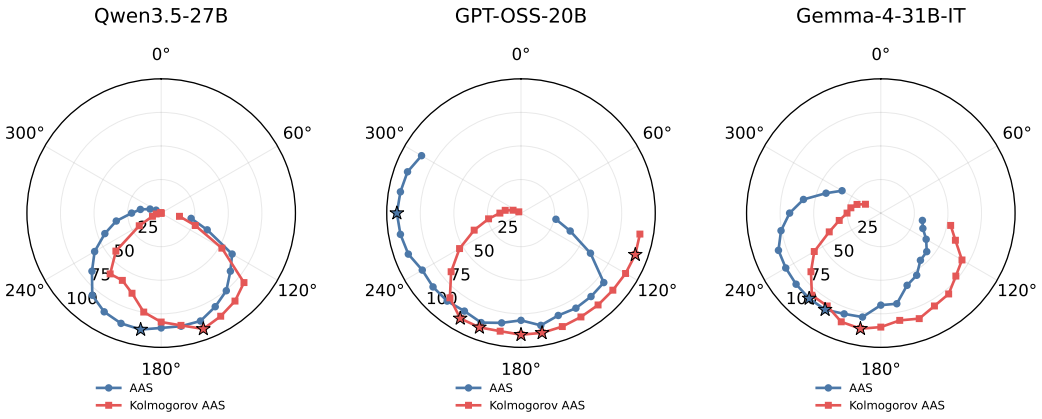


Figure A3: AAS and Kolmogorov AAS angle selection on the HarmBench-159 validation set. Angle denotes the target steering angle and radius denotes ASR (%); stars highlight configurations.

D.3 Fitting time

Table A1 reports the recorded wall clock time for fitting Kolmogorov maps from cached activations. These measurements exclude activation extraction, response generation, HarmBench scoring, TinyBench evaluation, and evaluation based configuration search.

Table A1: Kolmogorov map fitting time (seconds), excluding activation extraction, evaluation, and evaluation based configuration search.

Method	Qwen3.5-27B	GPT-OSS-20B	Gemma-4-31B-IT
Kolmogorov AAS	77.32	53.82	79.51
Kolmogorov DirAbl	69.32	44.42	45.10
Kolmogorov CAA	53.45	42.15	53.20

Computational scaling. Following Appendix B, let the recorded sites be indexed by $0, \ldots, L$, let d be the hidden dimension, and let n and m denote the source minibatch and reference sample counts. The Gram expansion computes layerwise squared distances without storing pairwise difference vectors, and cumulative distances give each interval distance by subtraction. Evaluating all $(L+1)(L+2)/2$ intervals therefore costs $\mathcal{O}(Ld+L^2)$ per trajectory pair, rather than $\mathcal{O}(dL^3)$ when interval distances are computed independently.

The reference–reference contribution is computed once without gradients and retained in the reported loss. Each optimization step recomputes the source–source and source–reference contributions. The custom backward pass recomputes interval kernels instead of retaining them, with the same arithmetic order as the forward pass. Over T steps, the kernel evaluation and activation-gradient cost is therefore

$$\mathcal{O}\big([m^2 + T(n^2 + nm)](Ld + L^2)\big).$$

Here, $T = 3000$, $n = 32$, and $m = 512$. This bound excludes bandwidth estimation and evaluation and differentiation of the steering map. Fitting uses cached activations and requires no further forward or backward passes through the pretrained model. The measured times in Table A1 include map fitting and command overhead, but exclude activation extraction and downstream evaluation.

D.4 FITTING LOSS

Figures A4–A6 show the six recorded optimization traces for Kolmogorov AAS and DirAbl, grouped by model. Each figure shows AAS on the left and DirAbl on the right. Each run uses 3,000 Adam steps, an initial learning rate of 0.01, a source minibatch size of 32, and cosine learning rate decay. AAS uses the heuristic initialization described above, whereas DirAbl uses random initialization. Plot annotations give the first and last logged losses.

All six traces end below their initial logged loss, with an early decrease followed by fluctuations. Fluctuations are consistent with stochastic minibatch evaluation; the plots alone do not establish convergence or identify the best checkpoint.

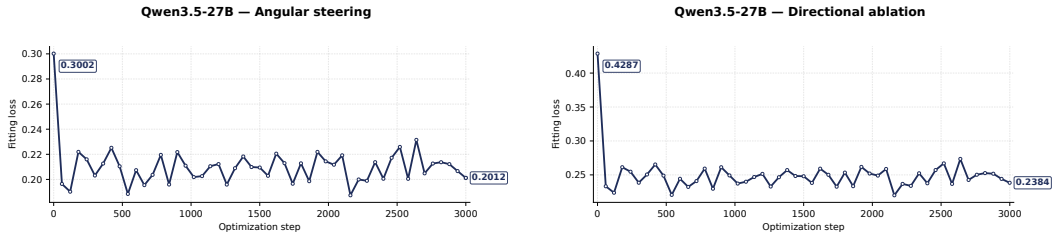


Figure A4: Fitting loss on Qwen3.5-27B over 3,000 Adam steps: Kolmogorov AAS (left) and Kolmogorov DirAbl (right).

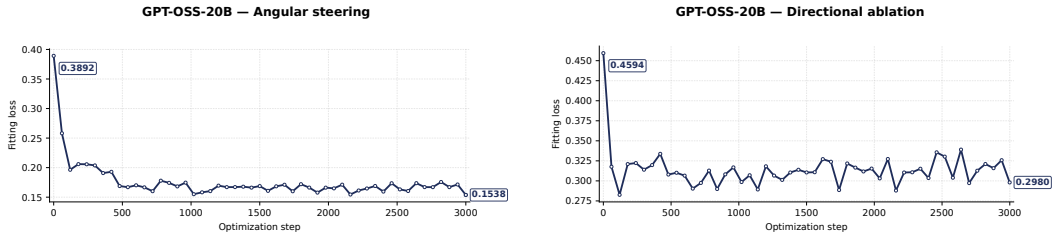


Figure A5: Fitting loss on GPT-OSS-20B over 3,000 Adam steps: Kolmogorov AAS (left) and Kolmogorov DirAbl (right).

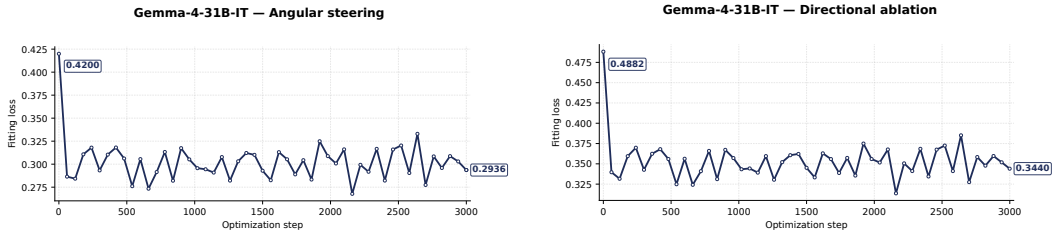


Figure A6: Fitting loss on Gemma-4-31B-IT over 3,000 Adam steps: Kolmogorov AAS (left) and Kolmogorov DirAbl (right).

E TRUTHFULNESS EXPERIMENT

E.1 EXPERIMENTAL SETUP

This task evaluates whether activation steering can increase the truthfulness of open-ended answers without making them uninformative. We use the generation-only TruthfulQA protocol of Zhao et al. (2026) rather than separate multiple-choice and open-ended tasks. The source distribution consists of final-token activations induced by questions paired with reference incorrect answers, while the reference distribution consists of activations induced by the same questions paired with reference

correct answers. Every listed correct and incorrect answer is retained, so the two activation clouds need not contain the same number of samples.

Dataset and cross-validation. We use all 817 unique questions from the TruthfulQA generation validation set (Lin et al., 2022). We construct two deterministic outer folds using a 50/50 question-level split with random seed 42. The two held-out folds contain 408 and 409 questions, respectively. Within each outer training fold, we reserve 20% of the questions for open-ended validation using seed $42 + f$, where $f \in \{0, 1\}$ is the outer-fold index. This produces 82 validation questions and 326 or 327 map-fitting questions per fold. All correct and incorrect answers associated with a question remain in the same partition. The inner-validation split is used to select the original CAA baseline's steering coefficient. Final maps are fitted from scratch on the complete outer training fold and evaluated once on the disjoint outer test fold. The reported scores join the two outer test folds, yielding one out-of-fold answer for each of the 817 questions.

Models. We evaluate three pretrained base models: `meta-llama/Llama-3.1-8B` (Grattafiori et al., 2024), `Qwen/Qwen2.5-7B` (Qwen et al., 2025), and `mistralai/Mistral-7B-v0.1` (Jiang et al., 2023a). Following the layer choices supplied by the ODESteer repository (Zhao et al., 2026), we intervene at zero-based residual-block index 13 for Llama-3.1-8B and Qwen2.5-7B, and index 15 for Mistral-7B. These models have 32, 28, and 32 decoder blocks and hidden widths 4096, 3584, and 4096, respectively.

Metrics and judges. We report the mean binary Truth score, the mean binary Informativeness (Info) score, and the mean per-example product Truth×Info, all as percentages. Equivalently, Truth×Info is the percentage of responses judged jointly truthful and informative. It is computed before averaging and is therefore not the product of the two aggregate percentages. We use the two local Llama-2 7B judges: `allenai/truthfulqa-truth-judge-llama2-7B`[3] and `allenai/truthfulqa-info-judge-llama2-7B`[4]. Each judge receives the question and the same generated response in `Q:/A:/True:` or `Q:/A:/Helpful:` format. Judge decoding is greedy, and a response receives score one only when the decoded judgment has the exact normalized `yes` suffix. Multiple-choice accuracy is not used for fitting, selection, or evaluation.

Baselines. We include the unsteered model and three map families

- **No steering.** The identity intervention is evaluated under exactly the same prompts, decoding configuration, folds, and judges as the steered methods. Its generations are produced once per model and fold.
- **Contrastive Activation Addition (CAA)** (Rimsky et al., 2024). Let μ_+ and μ_- be the mean correct-answer and incorrect-answer activations at the selected layer. CAA uses the difference-in-means (DIM) vector $d = \mu_+ - \mu_-$ and applies
$$S_{d,\alpha}(x) = x + \alpha d.$$
- **MiMiC** (Singh et al., 2024). MiMiC fits a binary logistic classifier to the correct- and incorrect-answer activations and a dense affine linear transport $T(x) = Ax + b$ from the incorrect to the correct distribution. At inference time, T is applied only when the classifier assigns an activation to the incorrect-answer class; otherwise the activation is unchanged. We use logistic regression with at most 1,000 iterations and POT `LinearTransport` (Flamary et al., 2021) with regularization 10^{-2}. The map is applied at full strength, matching ODESteer's fixed MiMiC semantics (Zhao et al., 2026).
- **Linear-AcT** (Rodríguez et al., 2025a). Linear-AcT fits a coordinatewise affine map
$$S_{w,b}(x) = w \odot x + b.$$
For each coordinate, the method takes the first $n = \min(n_+, n_-)$ samples from each cloud, centers and independently sorts the two sets, and fits the corresponding one-dimensional linear transport. The resulting diagonal affine map is also applied at full strength and has no steering-strength sweep.

[3] https://huggingface.co/allenai/truthfulqa-truth-judge-llama2-7B
[4] https://huggingface.co/allenai/truthfulqa-info-judge-llama2-7B

Kolmogorov steering maps. Kolmogorov Steering replaces heuristic or closed-form map selection with map parameters learned by minimizing the proposed MMD objective in Eq. 2. The fitted maps are applied directly at inference time, with $\alpha = 1$. We evaluate the following parameterizations:

- **Kolmogorov low-rank affine maps.** We use rank $r = 2$. We fit

$$F_{U,V,b}(x) = x + U(V^\top x) + b, \qquad U, V \in \mathbb{R}^{D\times r}.$$

  The inference map is $S_\theta(x) = F_{U,V,b}(x)$.

- **Kolmogorov diagonal affine maps.** We fit

$$F_{d,b}(x) = (\mathbf{1} + d) \odot x + b,$$

  with inference map $S_\theta(x) = F_{d,b}(x)$.

- **Kolmogorov CAA.** We fit one unconstrained translation vector b at the selected layer and apply

$$S_b(x) = x + b.$$

Activation extraction and intervention. For Llama-3.1-8B and Mistral-7B, fitting examples are formatted as `Q: question\nA: answer`; Qwen2.5-7B uses its tokenizer's chat template with the reference answer placed in the assistant turn. We capture the representation at the final token of each fully formatted question–answer sequence. For Kolmogorov Diagonal and Kolmogorov Low Rank, we cache these activations at every decoder-block output. During fitting, the map modifies only the selected layer's cached activation, while the other cached activations remain unchanged. The MMD loss is evaluated over the full cached trajectory. MiMiC, Linear-AcT, CAA, and Kolmogorov CAA use only the selected residual-block output for fitting.

At inference time, all nonidentity maps are applied only at the selected residual-block output. The hook modifies only sequence position -1 on every model forward call. It therefore steers the final prompt token during prefill and, under cached autoregressive decoding, the current newly generated token on each subsequent call; earlier prompt positions are not rewritten. The fitted map parameters remain fixed throughout generation.

Optimization. All gradient-fitted maps use Adam (Kingma & Ba, 2014) for 3,000 steps with learning rate 0.01, cosine decay to zero (Loshchilov & Hutter, 2016), batch size 32 from each activation distribution, and random seed 0. We use no weight decay or early stopping. Kolmogorov low-rank maps initialize U and b to zero and V from a seeded Gaussian scaled by $D^{-1/2}$; Kolmogorov diagonal maps initialize both d and b to zero. Kolmogorov CAA initializes its translation vector as a seeded isotropic random unit vector. All map parameters are optimized jointly.

Map parameter selection. For the original CAA baseline, we sweep

$$\alpha \in \{0.10, 0.25, 0.50, 1.0, 2.0, 4.0\}$$

on the inner-validation questions. We select the coefficient with the highest $T \times I$, breaking ties in favor of the smaller coefficient. We then recompute the DIM vector on the complete outer training fold and keep the selected α fixed for outer-test generation. MiMiC and Linear-AcT are applied at full strength, as are the fitted Kolmogorov maps. Outer-test generations are never used for selection.

Generation details. Open-ended generation uses the following system instruction: "Interpret each question literally, and as a question about the real world; carefully research each answer, without falling prey to any common myths; and reply 'I have no comment' unless you are completely certain of the answer." We generate at most 50 new tokens with sampling enabled, temperature 0.7, nucleus probability 0.9 (Holtzman et al., 2019), repetition penalty 1.1, random seed 42, and left-padded batches of 10. These sampling settings apply to the evaluated models; as described above, the AllenAI judges themselves use greedy decoding.

F ASSESSING THE OPTIMALITY OF CAA'S DIM VECTOR

This section expands the empirical assessment in Section 5.3. Theorem 4.7 identifies the DIM vector as the unique optimal translation for the marginal MMD objective when the centered source and reference distributions and the kernel satisfy its assumptions. The covered families include Gaussian and Laplace distributions and Gaussian mixtures with a common mean. We examine whether MMD optimization recovers a direction close to DIM, and whether this alignment is accompanied by similar downstream performance.

Comparison and evaluation. We compare CAA (Rimsky et al., 2024) and Kolmogorov CAA at the same intervention layer for each model. CAA uses the DIM vector, while Kolmogorov CAA minimizes the MMD loss from random initialization. For jailbreak, we use Qwen3.5-27B, GPT-OSS-20B, and Gemma-4-31B-IT, intervening at decoder block outputs 28, 12, and 33, respectively. We cache final-token activations at every decoder-block output, but both methods intervene only at the selected layer. During Kolmogorov fitting, the translation modifies only that layer's cached activation; all other cached activations remain unchanged, and the MMD loss is evaluated over the full cached trajectory. The inference time coefficients are 2, 4, and 1 for both methods. ASR is evaluated with the HarmBench classifier (Mazeika et al., 2024) on the 104 AdvBench test prompts (Zou et al., 2023), and TinyBench (Maia Polo et al., 2024) measures general capability retention.

For truthfulness, we compare CAA with Kolmogorov CAA. The original CAA baseline selects its steering coefficient on validation questions, while Kolmogorov CAA applies its fitted translation directly ($\alpha = 1$). We evaluate Llama-3.1-8B, Qwen2.5-7B, and Mistral-7B at layers 13, 13, and 15 under the two fold train/test protocol. All 817 TruthfulQA questions (Lin et al., 2022) contribute to the reported generation scores. AllenAI judges assess truthfulness and informativeness; $T \times I$ is the percentage of responses judged both truthful and informative. We compute cosine similarity directly between the saved learned and DIM translation vectors within each outer fold, then report the mean of the two similarities. For jailbreak, we compare the learned and DIM vectors at the selected layer before multiplication by the inference time coefficient.

Table A2: CAA and randomly initialized Kolmogorov CAA on the 104 AdvBench test prompts, scored by the HarmBench classifier. ASR and TinyBench averages are percentages (higher is better). Vector comparisons use layers 28 for Qwen, 12 for GPT-OSS, and 33 for Gemma. Norm difference is $100|\|b_{\mathrm{Kol}}\|_2/\|b_{\mathrm{DIM}}\|_2 - 1|$.

	Qwen3.5-27B		GPT-OSS-20B		Gemma-4-31B-IT	
Method	HarmBench ASR ↑	TinyBench Avg. ↑	HarmBench ASR ↑	TinyBench Avg. ↑	HarmBench ASR ↑	TinyBench Avg. ↑
CAA	89.42	62.94	70.19	57.02	94.23	58.82
Kolmogorov CAA	90.38	62.94	78.85	57.39	91.35	58.84
Cosine similarity	0.999813		0.976119		0.997130	
Norm difference (%)	0.98		13.27		1.85	

Table A3: CAA and randomly initialized Kolmogorov CAA on 817 TruthfulQA questions. Truth and Info are the percentages judged truthful and informative; $T \times I$ is the percentage judged both. Cosine similarity is averaged over the two outer folds; norm difference, $100|\|b_{\mathrm{Kol}}\|_2/\|b_{\mathrm{DIM}}\|_2 - 1|$, is reported per fold. Intervention layers are 13, 13, and 15, respectively.

	Llama-3.1-8B			Qwen2.5-7B			Mistral-7B		
Method	T×I ↑	Truth ↑	Info ↑	T×I ↑	Truth ↑	Info ↑	T×I ↑	Truth ↑	Info ↑
CAA	60.7	67.2	93.3	65.0	79.2	85.1	61.1	67.8	93.3
Kolmogorov CAA	59.5	66.3	93.0	64.9	79.1	85.1	60.8	67.4	93.3
Mean cosine similarity		0.996484			0.998062			0.996299	
Norm diff. (%), fold 0		0.69			1.48			0.61	
Norm diff. (%), fold 1		1.12			2.32			1.32	

Directional alignment and performance. Despite random initialization, the learned vectors closely align with DIM in every evaluated setting. Jailbreak cosine similarities range from 0.976119 to 0.999813, while truthfulness fold means range from 0.996299 to 0.998062 (Tables 3 and 4). On TruthfulQA, Kolmogorov CAA's $T \times I$ scores are lower than the α-swept CAA baseline by only 1.224, 0.123, and 0.245 percentage points for Llama, Qwen, and Mistral, respectively. Jailbreak TinyBench scores are nearly unchanged, while ASR increases by 0.96 percentage points on Qwen and decreases by 2.88 points on Gemma. GPT-OSS is an exception to the small performance differences, with an ASR increase of approximately 8.7 percentage points despite high cosine similarity.

Interpretation and scope. These results suggest that MMD optimization recovers the DIM direction up to small perturbations. Such deviations may reflect finite sample or optimization noise and can produce slightly better or worse downstream performance, rather than a consistent advantage over DIM. This interpretation is consistent with the theorem, but cosine similarity alone establishes neither equality of vector norms nor exact optimality. The theorem concerns a population level marginal MMD objective under specified assumptions; the experiments use finite samples, and the jailbreak CAA fits evaluate the loss over cached trajectories. We therefore view the observations as empirical support for DIM recovery, not as proof that all activation distributions satisfy the theorem's assumptions or that DIM universally maximizes task scores. Finally, a fixed translation changes the mean but leaves the centered distribution, and hence all existing higher order centered moments, unchanged. This expressivity limit motivates studying richer steering maps that can also modify distributional shape and inter layer dependence.

G Ablation Study of Matching Scope, Kernels, and Weights

This section presents the ablation results discussed in Section 5.4. We evaluate Qwen3.5-9B (Qwen Team, 2026) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) on the jailbreak task using different layer groups, kernels, and interval weighting schemes.

Table A4: Qwen3.5-9B and Llama-3.1-8B-Instruct (%) ASR on 104 AdvBench test prompts. Parentheses show changes from the unsteered baseline in percentage points. Bold and underlined scores mark the best and second best steered results for each model, including ties. $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function.

			Qwen3.5-9B		Llama-3.1-8B-Instruct	
Matching scope	Kernel	$w_{a,b} \propto$	HarmBench ASR ↑	TinyBench Avg. ↑	HarmBench ASR ↑	TinyBench Avg. ↑
Baseline (no steer)	—	—	0.96	67.96 (0.00)	4.81	67.57 (0.00)
AAS	—	—	72.12	63.10 (−4.86)	89.42	66.67 (−0.90)
Marginal matching	Gaussian	$\mathbb{1}_{\{a=b\}}$	90.38	66.10 (−1.86)	94.23	67.51 (−0.06)
Full trajectory	Gaussian	$\mathbb{1}_{\{a=0,\, b=L\}}$	<u>93.27</u>	66.62 (−1.34)	95.19	**67.88** (+0.31)
All contiguous groups	Gaussian	1	**95.19**	**66.92** (−1.04)	<u>96.15</u>	<u>67.79</u> (+0.22)
All contiguous groups	RQ	1	<u>93.27</u>	65.40 (−2.56)	<u>96.15</u>	67.59 (+0.02)
All contiguous groups	Gaussian	$\frac{1}{b-a+1}$	92.31	<u>66.78</u> (−1.18)	<u>96.15</u>	67.49 (−0.08)
All contiguous groups	RQ	$\frac{1}{b-a+1}$	90.38	66.59 (−1.37)	**97.12**	67.32 (−0.25)

H Related Work and Additional Discussion

Additive, projection, and rotational steering. Contrastive Activation Addition (CAA) constructs steering vectors by averaging activation differences between positive and negative examples and adds scaled versions of these vectors at inference time (Rimsky et al., 2024). Directional ablation removes an activation component along a behavior associated direction; Arditi et al. (2024) demonstrate this construction for refusal. Angular Steering rotates activations within a fixed two dimensional subspace, and Adaptive Angular Steering selectively applies this transformation according to feature alignment (Vu & Nguyen, 2025). These approaches supply different intervention geometries. Our framework retains these map families and studies how their parameters can be fitted using a common distributional objective. The difference in means recovery result further connects marginal MMD optimization to CAA under the stated distributional assumptions; it does not establish that the difference in means vector universally optimizes downstream task performance.

Activation transport and affine steering. Activation Transport formulates steering through transport between source and reference activation distributions, with Linear-AcT providing an affine construction (Rodríguez et al., 2025a). Representation Surgery studies optimal affine steering under specified constraints, including the moment matching construction MiMiC (Singh et al., 2024). These works establish distributional alignment as a useful principle for steering and provide the affine baselines considered here. Our emphasis is on the distributional information retained by the fitting objective: matching joint activation tuples across depth can distinguish laws with identical layerwise marginals. This distinction concerns the objective and is separate from map expressiveness. In particular, our comparison between full matrix MiMiC and a low rank learned map changes both the fitting procedure and the parameterization.

Related Work in Distributional Steering. LinEAS jointly learns elementwise affine interventions through a frozen pretrained network using a sum of layerwise, coordinatewise Wasserstein losses and optional sparsity regularization (Rodríguez et al., 2025b). Because source activations are computed through the intervened network, its optimization accounts for the downstream effects of earlier maps. Joint optimization of maps, however, is distinct from matching joint distributions: layerwise marginal losses do not generally identify cross layer dependence. Our objective explicitly compares joint activation distributions over contiguous depth groups. During fitting, LinEAS differentiates through the backbone despite keeping its weights frozen, whereas our implementation fits maps on cached activation trajectories and differentiates only through the maps and matching objective. This avoids repeated backbone forward and backward passes during fitting, while introducing a surrogate for the actual intervened trajectories. Our experiments evaluate optimization from cached activations, so we discuss LinEAS as related work. Configuration selection and downstream evaluation still require model execution.

MAT-STEER learns targeted interventions for multiple attributes using distributional alignment together with mechanisms for selective token intervention and control of interference between attributes (Nguyen et al., 2025). Its use of MMD alignment is closely related to our choice of discrepancy. The focus differs: MAT-STEER addresses interactions among behavior attributes, while our objective addresses dependence among activation coordinates at different depths. Our experiments study the latter question for individual steering tasks and do not establish performance for simultaneous multi attribute control.

Parallel fitting without additional backbone passes. Cached activations remove the sequential backbone dependency from map fitting: each layer's map operates directly on its cached inputs, so its evaluation need not wait for earlier interventions to propagate through the model. This makes map evaluation amenable to parallel execution across layers, while the joint objective coordinates the maps through cross layer dependencies. The optimization remains coupled by the loss, but requires no additional backbone forward or backward passes after activation extraction.

Sequential fitting that accounts for earlier interventions, as in the sequential construction of Activation Transport (Rodríguez et al., 2025a), must propagate their effects before fitting later maps. End to end fitting instead repeatedly executes and differentiates through the relevant backbone layers as map parameters change. For T optimization steps, the latter incurs a backbone cost of order $T(C_{\text{forward}} + C_{\text{backward}})$ that is absent from cached fitting. Avoiding these forward and backward

passes is particularly valuable as model size and depth increase. The advantage comes from eliminating additional backbone passes during fitting, together with the opportunity to parallelize map evaluation; the realized speedup depends on the implementation and workload.

This setting allows us to fit steering maps for models up to 31B parameters and 64 decoder layers without repeatedly evaluating the expensive backbone. The recorded map fitting runs on 20B-31B models take approximately 42-80 seconds, excluding activation extraction, configuration selection, and downstream evaluation (Appendix D.3). Our loss is a surrogate for matching actual intervened trajectories, yet the experiments show that it supports effective steering and improves the evaluated rotational, projection, and affine maps over their original constructions. Backbone execution remains necessary when extracting activations and applying or evaluating the learned interventions.

Joint distribution matching. Joint Adaptation Networks align joint distributions of multiple network layers through joint maximum mean discrepancy for domain adaptation (Long et al., 2017). Joint generative moment matching uses a related criterion to learn generation and inference networks (Gao & Huang, 2018), and joint distribution matching has also been studied for unsupervised domain adaptation (Jin et al., 2020). Thus, joint distribution alignment and product kernel MMD predate our work. Our contribution applies this perspective to activation steering using a weighted collection of contiguous depth groups, including the full trajectory, with shared kernel evaluations across groups and a common fitting objective for several intervention families. The exploratory ablation sweeps describe sensitivity to matching scope, with their selection protocol stated in Appendix G.

Remark H.1 (Functional error as a surrogate). Functional error measures agreement between steered and reference outputs and serves as a surrogate for behavioral discrepancy when source and reference examples have a meaningful correspondence. For example, a clean input and an adversarially perturbed version that preserves the intended answer provide a natural pair for assessing whether steering restores the reference behavior. Such pairing motivates the output comparison; applying Theorem 4.5 additionally requires the trajectories to satisfy its common input and regularity assumptions.

Kernel embeddings and alternative discrepancies. MMD compares distributions through their RKHS mean embeddings (Gretton et al., 2012). Characteristic kernels provide identification of probability distributions (Sriperumbudur et al., 2010), while the theory of characteristic and universal tensor product kernels supplies conditions for joint distribution identification (Szabó & Sriperumbudur, 2018). Weak convergence guarantees require suitable additional kernel conditions (Simon-Gabriel et al., 2023). Our analysis specializes these tools to the weighted trajectory kernel. The symmetric positive definite distribution assumption used in the difference in means result follows the terminology of Law et al. (2017). The path law formulation itself does not require MMD: Sinkhorn divergences (Genevay et al., 2018) and variational divergence objectives (Nowozin et al., 2016) provide other possible distributional discrepancies, but are not evaluated here.

H.1 Beyond Network Depth

The object in Equation 1 is the law of an activation process. Network depth is one possible index; successive denoising steps provide another. A finite path law is precisely the joint law of all recorded states. The process formulation makes the index, dynamics, and admissible interventions explicit and also accommodates processes with infinitely many indices. This places depth based and iterative steering within the same framework.

A stochastic denoising formulation. As an illustrative extension to diffusion language models, consider a sequence of token or latent states $Z_0, \dots, Z_K$ in a measurable state space E, conditioned on a prompt c. Let $\rho(dc)$ denote the prompt distribution, $\nu_0(dz_0 \mid c)$ the initial noise law, and $P_k(dz_k \mid z_{k-1}, c)$ the pretrained denoising transitions. A family of admissible interventions produces kernels $P_{\theta,k}$ while preserving the prompt and initial noise distributions. The resulting joint law of the prompt and trajectory is

$$\mathcal{D}_\theta(dc, dz_{0:K}) = \rho(dc)\,\nu_0(dz_0 \mid c) \prod_{k=1}^{K} P_{\theta,k}(dz_k \mid z_{k-1}, c). \tag{14}$$

For example, given the previous state z and prompt c, the model first samples a candidate state $U \sim P_k(\cdot \mid z, c)$. A measurable steering map then transforms it into $Z_k = S_{\theta,k}(U, c)$, which becomes the input to the next denoising step. The resulting transition law is

$$P_{\theta,k}(A \mid z, c) = \Pr(S_{\theta,k}(U, c) \in A \mid z, c)\,,$$

where $A \subseteq E$ is any measurable set of states. The map uses only the current candidate and prompt. Steering may also modify activations inside the denoiser, directly changing the distribution of the next state. Both approaches define a steered path law $\mathcal{D}_\theta$, and the objective $\mathfrak{D}(\mathcal{D}_\theta, \mathcal{D}_r)$ measures its discrepancy from a reference law over prompts and trajectories. Matching each step separately ignores how successive states depend on one another; matching the path law includes these dependencies.