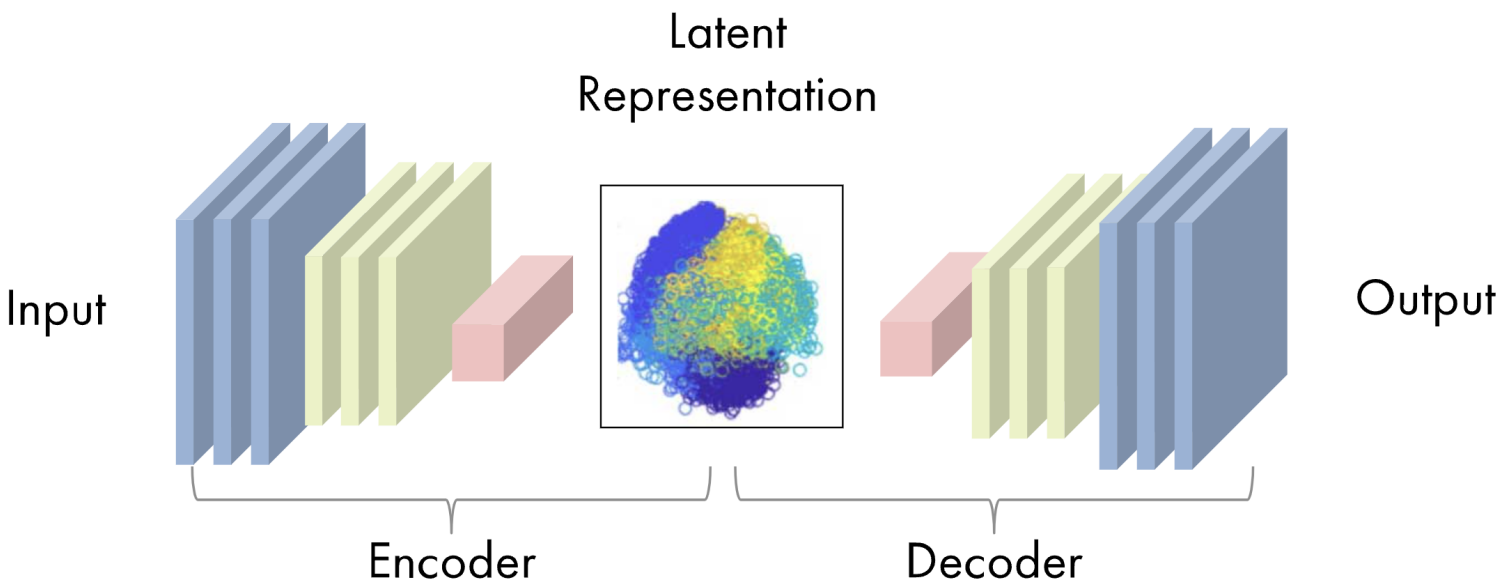
Causal Abstraction and Manifold Steering

Deven Misra

CD3 Hack Friday, May 2026

Table of Contents

Autoencoders



cms-ml

Constraining Representations

A. Gagliano and V. A. Villar, *A Physics-Informed Variational Autoencoder for Rapid Galaxy Inference and Anomaly Detection*, Dec. 2023

- Use loss function to force latent dimensions to correspond to orientation, redshift, stellar mass, and star-formation rate.

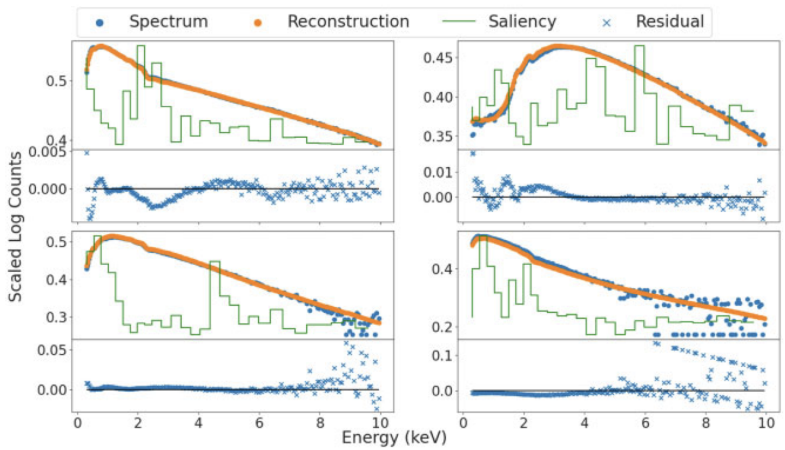
E. Tregidga *et. al*, *Rapid spectral parameter prediction for black hole X-ray binaries using physicalized autoencoders*, Mar. 2024

- Train decoder on labeled synthetic data (supervised learning)
- Train encoder with unlabeled observations and the trained decoder

Parameter Saliency

Table 3. Saliency calculated from the differentiation of the decoder's loss function with respect to each free spectral parameter averaged over all validation spectra (S_μ) and the standard deviation of the saliencies (S_σ) normalized by the smallest S_μ, as the absolute value is arbitrary.

	$N_{\rm H}$	Γ	$f_{\rm sc}$	$kT_{\rm disc}$	N
S_μ	2.97	1.00	1.22	8.92	8.83
S_σ	5.93	1.70	1.72	9.26	8.06



E. Tregidga *et. al*, *Rapid spectral parameter prediction for black hole X-ray binaries using physicalized autoencoders*, Mar. 2024

Table of Contents

Causal Abstraction

GOODFIRE

A framework for **mechanistic interpretability** — reverse-engineering the algorithms language models use internally using **causal abstraction**.

You write a high-level causal model describing *how you think* an LM solves a task, then run experiments to test whether the LM's internal components actually implement that algorithm.

goodfire-ai/causalab

A. Geiger *et. al*, *Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability*, May 2025

Core Concepts

Causal Models

A causal model is your hypothesis about how the LM solves a task. It consists of:

- **Variables**: concepts that might be represented in the network (e.g., "subject name", "indirect object")
- **Values**: possible assignments to each variable
- **Parent–Child Relationships**: directed dependencies
- **Mechanisms**: functions that compute a variable's value given its parents'

Causal Abstraction

Mechanistic interpretability aims to reverse-engineer the algorithm a network implements. Causal abstraction grounds this: an algorithm is a causal model, a network is a causal model, and "implementation" is the abstraction relation between two models. The algorithm is a **high-level causal model**, the network is a **low-level causal model**, and when the high-level mechanisms are accurate simplifications of the low-level mechanisms, the algorithm is a **causal abstraction** of the network.

Interchange Interventions

Interchange interventions test whether a high-level variable aligns with specific features in the LM. The intervention replaces activations from one input with activations from a counterfactual input, isolating one causal pathway at a time.

Method-level techniques for *constructing* the feature space being intervened on — DAS, DBM, PCA, Boundless DAS, SAE — live in `causalab/methods/` and are selected as options inside analyses (e.g. `subspace.method: das`, `locate.method: interchange`).

goodfire-ai/causalab

Table of Contents

Manifold Steering

We fit a smooth manifold within each space to the model's unintervened activations or outputs for a task: $\mathcal{M}_h \subseteq \mathcal{A}$, the *activation manifold*, and $\mathcal{M}_y \subseteq \mathcal{Y}$, the *behavior manifold*. To fit the activation manifold $\mathcal{M}_h$, we reduce activation vectors $\boldsymbol{h}(x)$ to 64 dimensions via PCA, compute "concept centroids" (e.g., averaging all activations where the correct answer is Wednesday), and fit cubic splines (Reinsch, 1967) through the centroids (see App. A.3 for further spline fitting details). To fit the behavior manifold $\mathcal{M}_y$, we follow a similar procedure but first map each centroid from the probability simplex onto Hellinger space via $p \mapsto \sqrt{p}$. This linearizes the geometry of the simplex: the Hellinger distance between distributions becomes an ordinary Euclidean distance, $d_H(p, q) = \frac{1}{\sqrt{2}}\|\sqrt{p} - \sqrt{q}\|$, so we can fit splines and compare distributions with standard Euclidean tools while still respecting the underlying probabilistic geometry (Amari & Nagaoka, 2000). Decoded points are squared back to recover valid distributions (further details, including how we keep the fit on the sphere, are in App. A.4). Unless stated otherwise, we use Llama 3.1 8B (Touvron et al., 2023) with activations from layer 28, and visualize manifolds via 3D PCA.

Manifold Steering

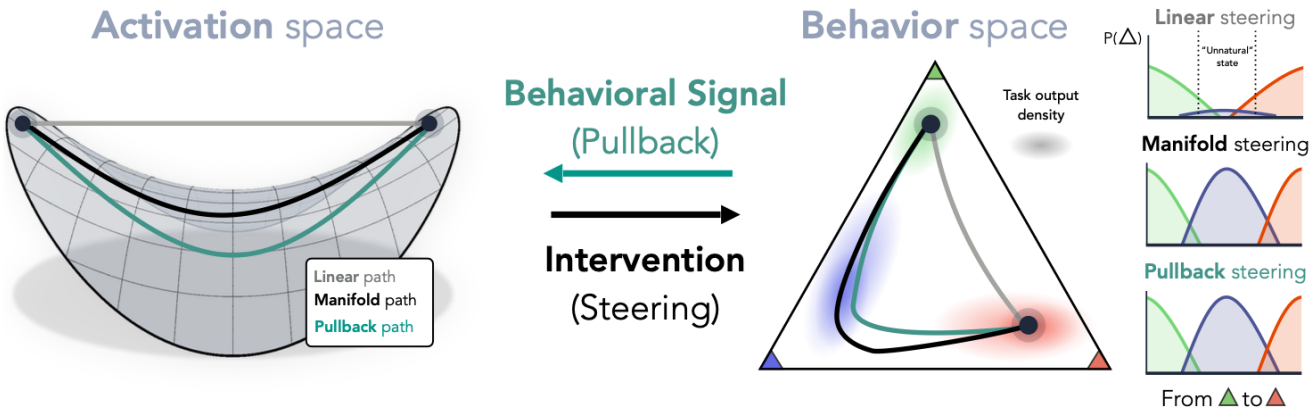


Figure 1: **How do different geometries of activation space modulate behavior?** We illustrate paths through activation space (left), each defined by a different geometry. Interventions along paths in activation space induce paths in behavior space (right, illustrated on a three-concept probability simplex). **Euclidean:** the standard approach of linear steering assumes a flat geometry and interventions follow a straight line. Such paths may cut across the activation manifold, yielding *unnatural* behavioral trajectories that pass through off-manifold regions of behavior space. **Density geometry:** a density-based metric whose geodesics follow the intrinsic geometry of a fitted activation manifold, yielding more natural transitions in behavior space. **Pullback geometry:** a behavior-aware metric obtained by "pulling back" behavior-space geometry into activation space, yielding paths that follow the manifold of natural (unintervened) output distributions. Overall, we argue that geometric structure in neural representations encodes the conceptual space a model is reasoning over, which in turn constrains its output behavior. Hence, manifolds in activation and behavior space are two images of the same underlying structure, and so we expect the density and pullback geometries to coincide.

Linear Steering vs. Manifold Steering

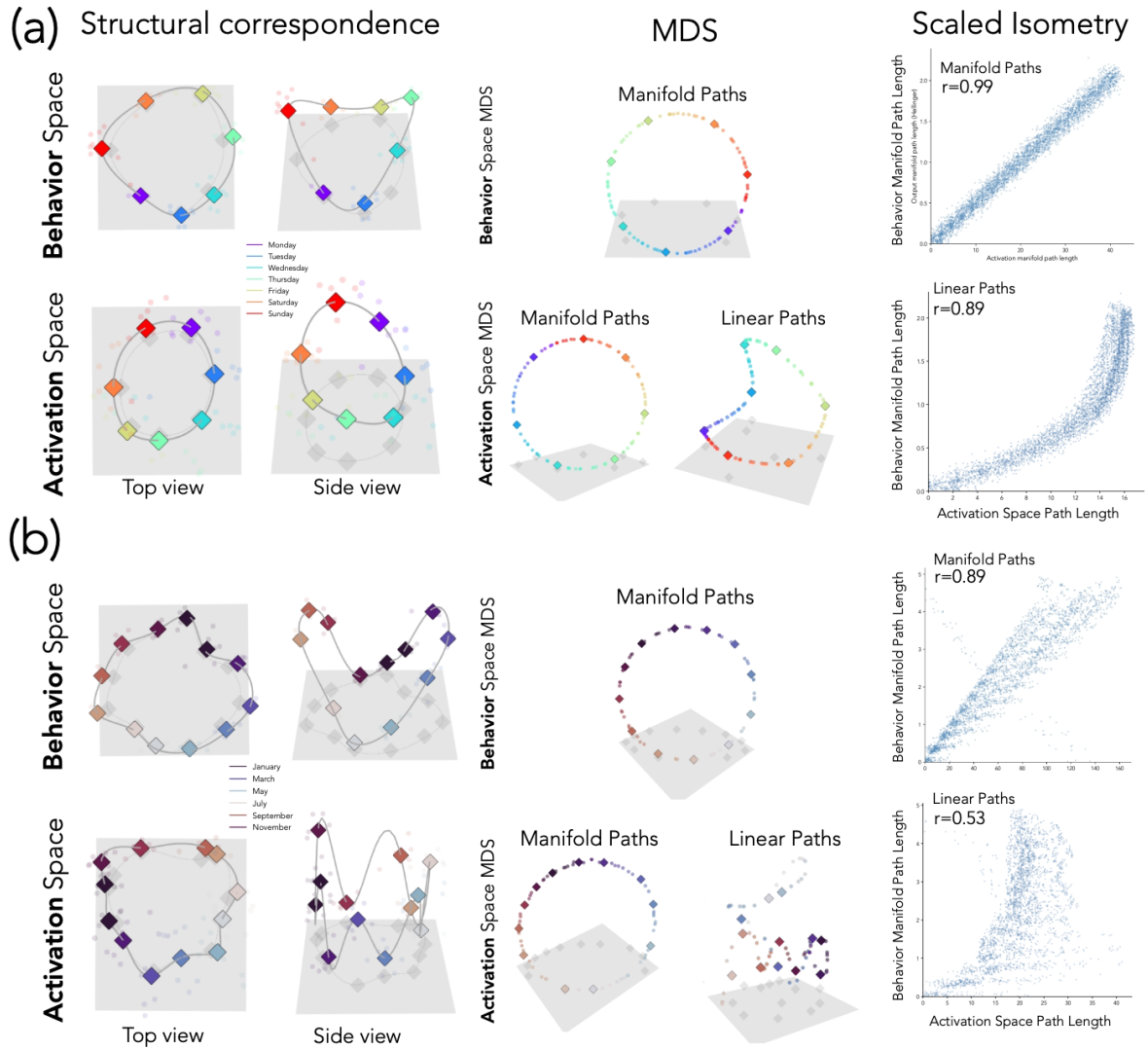
We consider two strategies, both constructed by interpolation between the endpoints; the strategies differ only in the coordinate system in which the interpolation is taken (Fig. 1):

$$\boldsymbol{\pi}_{\text{lin}}(t) = (1-t)\,\boldsymbol{h}_0^\star + t\,\boldsymbol{h}_1^\star \qquad \text{(linear steering);} \qquad (1)$$

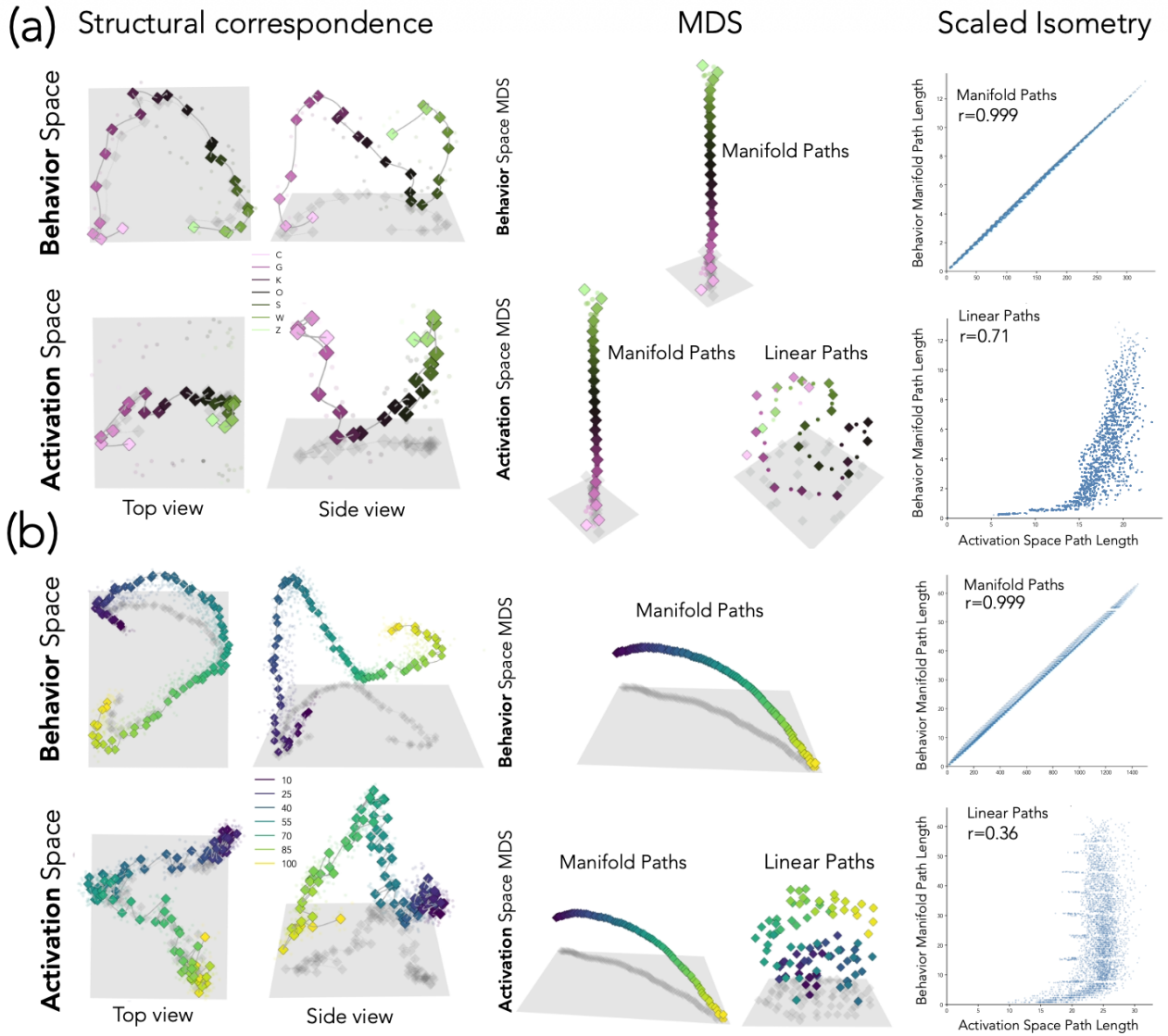
$$\boldsymbol{\pi}_{\text{m}}(t) = \boldsymbol{s}\big((1-t)\,\boldsymbol{u}_0 + t\,\boldsymbol{u}_1\big), \quad \boldsymbol{u}_i = \boldsymbol{s}^{-1}(\boldsymbol{h}_i^\star) \qquad \text{(manifold steering).} \qquad (2)$$

In the above, $\boldsymbol{s} : \mathbb{R}^k \to \mathcal{A}$ is a *parameterization* of $\mathcal{M}_h$—the map sending k-dimensional intrinsic coordinates to the corresponding point on the manifold in the activation space $\mathcal{A}$. Linear steering (also known as 'diff-in-means steering') (Bau et al., 2018; Subramani et al., 2022; Turner et al., 2023) interpolates in $\mathcal{A}$ directly—the standard additive-vector baseline. Manifold steering interpolates in the intrinsic coordinates of $\mathcal{M}_h$ and maps the result back through $\boldsymbol{s}$, so $\boldsymbol{\pi}_{\text{m}}$ stays on the activation manifold $\mathcal{M}_h$ throughout. Each strategy thus corresponds to a different choice of geometry on activation space, which we concretize in §3.4.

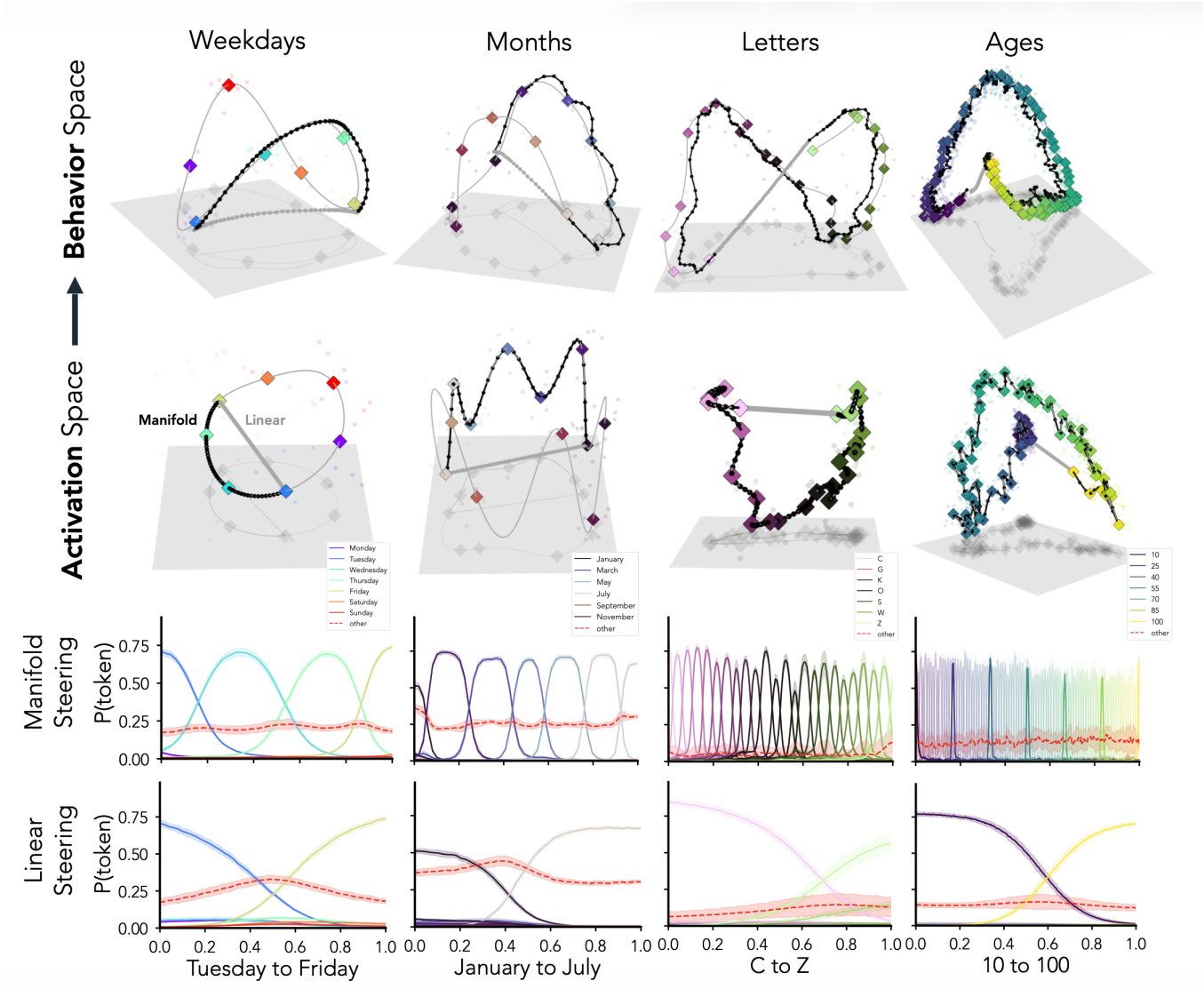
Cyclic Concepts



Sequential Concepts



Transition Behavior



Geometry

The Geometry of Steering: Consider a Riemannian metric $\boldsymbol{G}$, which assigns an inner product at each point of $\mathcal{A}$; together with a path $\boldsymbol{\pi} : [0, 1] \to \mathcal{A}$, this defines the notion of path length as follows.

$$L_{\boldsymbol{G}}(\boldsymbol{\pi}) = \int_0^1 \sqrt{\dot{\boldsymbol{\pi}}(t)^\top \boldsymbol{G}(\boldsymbol{\pi}(t))\, \dot{\boldsymbol{\pi}}(t)}\, dt. \tag{4}$$

Then, a geodesic is defined as the path of minimum length between two endpoints, and each choice of geometry picks out a steering strategy. The strategies of linear steering and manifold steering (§3.2), written as interpolations in two different coordinate systems (Eqs. 1, 2), are two such choices; the pullback procedure of §3.3 is a third. Now, we make all three geometries explicit.

Definition 1 (Geometries of Steering)**.** *Let $E : \mathcal{A} \to \mathbb{R}$ be an energy function such that $E(\boldsymbol{h}) \propto -\log p(\boldsymbol{h})$, and let $\boldsymbol{g}_y$ be a chosen Riemannian metric on $\mathcal{M}_y$. We define:*

$$\boldsymbol{G}_I = \boldsymbol{I}_n, \qquad \textit{(linear steering)} \tag{5}$$

$$\boldsymbol{G}_E(\boldsymbol{h}) = \left(\alpha\, e^{-E(\boldsymbol{h})} + \beta\right)^{-1} \boldsymbol{I}_n, \qquad \textit{(manifold steering)} \tag{6}$$

$$\boldsymbol{G}_F(\boldsymbol{h}) = \boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{h})^\top\, \boldsymbol{g}_y\big(\boldsymbol{F}(\boldsymbol{h})\big)\, \boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{h}) + \epsilon\, \boldsymbol{I}_n, \qquad \textit{(pullback)} \tag{7}$$

where $\alpha, \beta > 0$ are calibration constants, $\epsilon > 0$ regularizes the pullback, $\mathbf{F} : \mathcal{A} \to \mathcal{Y}$ is the function from naturally occurring activations to naturally occurring behaviors, and $\boldsymbol{g}_y$ is any Riemannian metric on $\mathcal{M}_y$ (e.g., the induced Hellinger metric used in our experiments).

Interpretation

- **The Flat Geometry G_I.** Linear steering treats activation space as Euclidean: all directions and regions are equally valid, with Geodesics as straight lines $\boldsymbol{\ell}(t) = (1-t)\,\boldsymbol{h}_0 + t\,\boldsymbol{h}_1$. This geometry thus encodes no knowledge of naturally occurring activation or outputs.
- **The Density Geometry G_E.** Manifold steering derives a geometry for activation space from naturally occurring internal representations. Specifically, consider the geometry induced from an energy function $E(\boldsymbol{h}) \propto -\log p(\boldsymbol{h})$ by rescaling the identity according to local density. Here $e^{-E(\boldsymbol{h})}$ plays the role of an unnormalized density: large where activations concentrate (on $\mathcal{M}_h$) and small where they are sparse (off $\mathcal{M}_h$). The inverse makes off-manifold regions expensive and on-manifold movement cheap, with constants $\alpha, \beta > 0$ calibrating the dynamic range (Béthune et al., 2025). Geodesics under $\boldsymbol{G}_E$ thus follow $\mathcal{M}_h$, recovering manifold steering.
- **The Pullback Geometry G_F.** The steering path given by pullback derives geometric structure from naturally occurring model outputs. Specifically, $\boldsymbol{G}_F$ is the pullback of a chosen geometry on $\mathcal{M}_y$ through the Jacobian of the map from activation space to behavior space $\boldsymbol{F} : \mathcal{A} \to \mathcal{Y}$. By construction, path length under $\boldsymbol{G}_F$ equals path length of the induced behavioral trajectory along $\mathcal{M}_y$ (up to a regularization term). Geodesics under $\boldsymbol{G}_F$ are therefore activation paths whose induced behavioral trajectories are geodesics on $\mathcal{M}_y$—exactly the pullback construction of §3.3. The regularization $\epsilon\,\boldsymbol{I}_n$ ensures positive definiteness, since $\boldsymbol{J_F}$ has rank at most $|\mathcal{Z}| - 1 \ll n$; as ϵ tends to 0, the geometry approaches the pure pullback in the range of $\boldsymbol{J_F}$ and remains Euclidean in its null space.

Overall, we claim that while the metrics $\boldsymbol{G}_E$ and $\boldsymbol{G}_F$ are derived from different sources (internal activations and outputs, respectively), they converge on approximately the same paths in activation space (§3.3). This suggests the manifolds $\mathcal{M}_h$ and $\mathcal{M}_y$ are two images of the same conceptual geometry, related by an approximate Riemannian isometry. Consequently, the question of optimally steering model behavior boils down to isolating the geometry of a concept and defining operators to navigate it.

Summary

- Manifold steering produces smooth and ordered behavioral transitions: probability mass shifts steadily through adjacent values of a concept.
- Linear steering instead exhibits 'teleportation': mass jumps between non-adjacent concepts as the straight line cuts through the manifold's interior.
- Interventions along the activation manifold $\mathcal{M}_h$ produce natural output trajectories that follow $\mathcal{M}_y$. Therefore, outputs produced under manifold steering resemble those produced without intervention.

D. Wurgaft *et. al*, *Manifold Steering Reveals the Shared Geometry of Neural Network Representation and Behavior*, May 2026