The Shape of Beliefs: Geometry, Dynamics, and Interventions along Representation Manifolds of Language Models' Posteriors

Anonymous ACL submission

Abstract

Large language models represent prompt-conditioned beliefs (posteriors over answers and claims), but we lack a mechanistic account of how these beliefs are encoded in representation space, how they update with new evidence, and how interventions reshape them. To make these questions measurable, we study a controlled numerical setting where `Llama-3.2` infers a parametric posterior predictive distribution from stochastic time series, yielding a map from hidden states to output distributions and forming curved "belief manifolds." This lets us follow belief updates as trajectories when the underlying data-generating process switches, and then test interventions that try to move the model's belief along a desired coordinate. We demonstrate that standard linear steering often pushes states off-manifold and induces coupled, out-of-distribution shifts, while geometry- and field-aware steering better preserves the intended belief family.

1 Introduction

AI agents act in the world by making implicit inferences about the current state of reality. Their processing core, large language models (LLMs), are conditional belief engines that continuously assign probabilities to claims – what we refer to here as *beliefs*. Formally, given an input prompt $\mathcal{I}$ and a factual claim c, a belief is the conditional probability:

$$P(c = \text{true} \mid \mathcal{I}).$$

These beliefs have multiple origins. Some are *priors* acquired over the course of training and fine-tuning (sometimes with engineered calibration, for instance to guard against harmful content or instill political stances) and persist robustly across inputs; for example: the Earth is round. Others, typically called *posteriors*, are formed and updated at inference time from the prompt's content (Minder

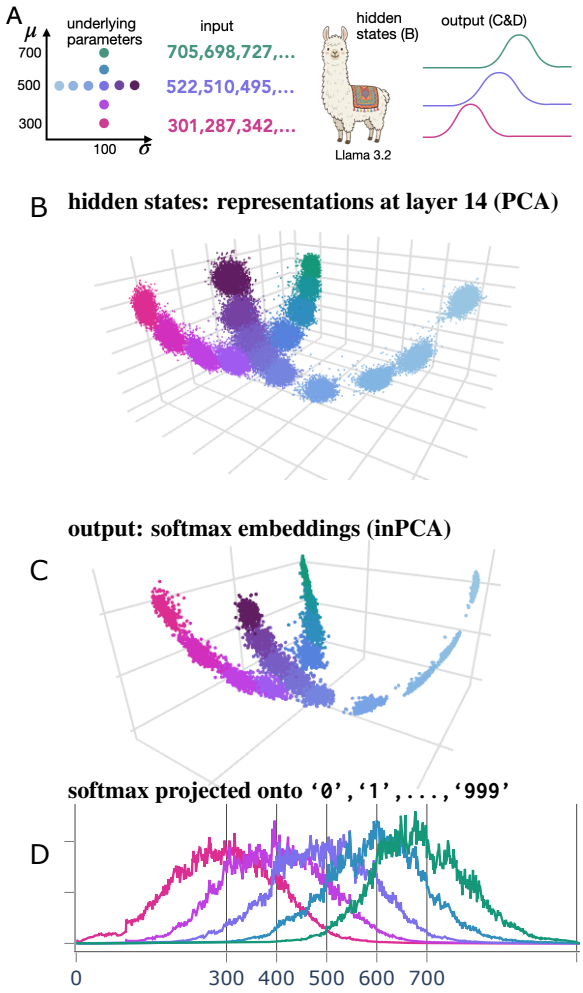

Figure 1: **The Shape of beliefs.** (A) Stochastic time series $u(t) \in [\![0, 999]\!]$ are generated from normal distributions $\mathcal{N}_{\mu,\sigma}$ and passed into `Llama 3.2` as strings. (B) Representations (PCA at layer 14) form curved manifolds parametrized by $(\mu, \sigma_0 = 100)$ (pink to green) and $(\mu_0 = 500, \sigma)$ (shades of purple). They encode the model's current posterior inferred from the input data. (C) Softmax outputs, visualized with inPCA, mirror the geometry of activations. (D) Softmax probabilities (posterior predictive), projected onto the subset of tokens corresponding to integers from 0 to 999, closely reflect the input distributions.

et al., 2025); for example: Alice is the character of a novel, or Alice is a famous writer, or Alice is a large ion collider experiment – depending on context. Finally, causal interventions such as steering aim to divert a model's belief from its "natural course," based on knowledge of how beliefs are represented internally (Bigelow et al., 2025).

Crucially, LLMs do not rely on isolated beliefs but on systems of beliefs: structured collections of interdependent claims whose probabilities evolve with context. Understanding how these belief systems are encoded, updated, and manipulated is essential for explaining model behavior. This paper takes a step toward that goal by studying belief systems as first-class objects of analysis, leading to three main questions:

- **Geometry:** How is a prompt-conditioned posterior belief encoded in the model's internal representations, and what geometric structure does it form?
- **Dynamics:** When evidence changes over time, how does the model update that posterior – do trajectories move along or across belief manifolds?
- **Interventions:** If we try to manipulate a belief by intervening on activations, when do standard linear methods succeed, and when do they produce unintended behavior?

Assessing a model's beliefs is often a difficult task, in particular because its behavior might not be readily inferrable from a model output's distribution (logits). That's why many studies rely on carefully designed prompts so that a model's belief can be read out directly from its verbal output (Bigelow et al., 2024). An alternative is to seek a signature of these beliefs in the internal representations, because representations encode not just a next-token prediction, but also contextual information and reasoning processes (Razzhigaev et al., 2025; Hu et al., 2025).

Our strategy is to construct a tractable belief family with ground truth and a continuous parametrization. We leverage the emergent ability of LLMs to infer distributions from numerical sequences: given a prompt containing a time series of numbers sampled from a generating distribution $\mathcal{G}_\theta$, the model converges to a next-token prediction that closely matches $\mathcal{G}_\theta$. This gives us a controlled notion of a posterior over a small set of interpretable parameters θ (e.g., $\theta = (\mu, \sigma)$ for Gaussians), and it lets us measure both outputs (logits/softmax) and their preimages (hidden states) in a calibrated way with ground-truth anchors.

Using `Llama-3.2` and prompts consisting of stringified integers, we tile the representation space with activation ensembles whose output distributions are approximately $\mathcal{N}(\mu, \sigma)$ over the integer tokens (0 to 999). The resulting preimages form smooth and curved manifolds parametrized by (μ, σ), providing a concrete geometric object that we can treat as a small, well-defined *conceptual space* in the sense of Gärdenfors (Gärdenfors, 2000): a continuous domain of related beliefs with interpretable coordinates. A belief system is not a single scalar or direction, but a structured region of representation space supporting families of related posteriors.

Once a geometric map of beliefs and their transitions is established, two additional steps follow. First, we seek a readout that describes how an interpretable belief coordinate, such as μ, can be decoded throughout the manifold; **we implement linear field probes** (Yocum et al., 2025), which characterize not just separability but the geometry of a probe family across a continuous domain. Second, we investigate intervening along that coordinate, by **comparing standard linear steering to geometry aware and field-aware steering schemes** that attempt to respect the manifold structure.

2 Experiments and analysis

Our goal is to make a posterior belief state both observable in the logits and indexable by a small set of interpretable parameters, so we can study its preimage geometry and update trajectories along this manifold. Stochastic numerical time series provide this setting: the model's next-token distribution can be compared directly to the ground-truth generating distribution $\mathcal{G}(\theta)$, and varying θ provides a continuous family of belief states.

Liu et al. (2024) have showed that LLMs are capable of various time series extrapolation tasks; in particular, given a prompt such as:

```
533,460,689,432,501,487,508,465,340,
```

where the numbers are random variables drawn from a normal distribution $\mathcal{N}\,(\mu = 500, \sigma = 100)$ (as an example), the model quickly in-context learns the distribution underlying the input series of (stringified) numbers, and reproduces this distribution in its logits (Appendix A).

For convenience, we rely primarily on the `Llama 3` models (Grattafiori et al., 2024), whose tokenizer discretely represents every single number between 0 and 999 (Bao et al., 2025). We restrict the input distribution to integers in that range by rounding and clamping. No finetuning or reinforcement learning is applied to open-source *base* models.

In our setup, there are two categories of tokens: the comma-predicting-number (`com2num`) tokens and the number-predicting-comma (`num2com`) tokens. Unless otherwise noted, we consider the `com2num` tokens in what follows. More detail in A.

2.1 Methods

We use Principal Component Analysis (PCA) to visualize and analyze high-dimensional sets of vectors. For vectors describing probability distributions, such as softmax outputs, we employ *Intensive PCA* (inPCA) instead (Quinn et al., 2019), a variation of PCA more informative for prediction vectors, i.e. constrained to the unit d-simplex.

2.2 Notations and definitions

We write vectors in bold (e.g., $\boldsymbol{v}$) and matrices in capital letters (e.g., $\boldsymbol{A}$). We use *activations*, *representations*, and *hidden states* interchangeably to refer to the model's residual stream, denoted $\boldsymbol{x}_{i,k}(l)$, where i is the token index, k the sequence index, and $l \geq 0$ the layer index.

A normal distribution with mean μ and standard deviation σ is written $\mathcal{N}(\mu, \sigma)$ or $\mathcal{N}_{\mu,\sigma}$. Its *preimage* manifold is denoted $\mathcal{M}_{\mu,\sigma}$: if $\boldsymbol{x} \in \mathcal{M}_{\mu,\sigma}$, after layer-norm and unembedding it maps to a softmax corresponding to $\mathcal{N}_{\mu,\sigma}$. For preimages at intermediate layers l (not the final layer), we write $\mathcal{M}^{(l)}_{\mu,\sigma}$.

Logits serve as a proxy for model behavior. To compare input and output distributions, we use probability vectors obtained by applying softmax to the logits with temperature $T = 1.0$.

3 Results

We first investigate the geometry of belief manifolds tiled by the activations from various input time series. Then, we report the dynamics along and across these manifolds when time series switch between distributions. Finally, we describe how the belief parametrization, materialized by μ, is encoded along the activation manifold using linear field probes, and evaluate causal interventions based on the discovered geometry.

3.1 Shape of beliefs

Given enough number samples, of the order of 100, `Llama 3.2` converges towards the true distribution underlying the data (Appendix A), as evidenced by its logit output in Fig. 1C & D. This is in-context learning (Brown et al., 2020). In other words, over the course of the prompt the model acquires a posterior belief about the input contextual information.

This belief is not only manifested in the logits, but also encoded in the internal representations of the prompt. We extract an ensemble of activations corresponding to the preimages for normal distributions with various means and standard deviations $\mathcal{N}_{\mu,\sigma}$. Fig. 1B reveals two orthogonal manifolds, one corresponding to varying means $\mu \in \{300, 350, \ldots, 700\}$ and constant standard deviation $\sigma_0 = 100$, the other to constant mean $\mu_0 = 500$ and varying $\sigma \in \{20, 50, \ldots, 200\}$. These manifolds appear to form smooth, continuous structures, exhibiting substantial curvature. This implies, *a priori*, that the geometry supporting the data is complex, and might not be adequately described by standard linear frameworks such as the Linear Representation Hypothesis (LRH, (Park et al., 2024)).

This tiling of a well-defined manifold can be interpreted as constituting a conceptual space (Gärdenfors, 2000; Hindupur et al., 2025).

3.2 Belief dynamics

Now that we have established a ground-truth map of beliefs, we investigate the situation where the input materializes a sharp change of distribution and the corresponding response of the model to this impulse perturbation. Concretely, we study the model's output distribution when we input a time series whose first 1000 numbers follow $\mathcal{N}(300, 100)$ and the next ones abruptly switch to $\mathcal{N}(700, 100)$ (Fig. 2A).

We observe two timescales of adjustment in Fig. 2B. The mean of the output distribution quickly settles on the new mean. The variance takes longer to equilibrate, defining an effective *timescale of belief equilibration*. The specific trajectory that the model follows is reflected both in probability space (Fig. 2D) and in representation space (Fig. 2C). They show that, after the switch, the model jumps to a phase of high variance, which can be seen equivalently as a *high entropy* state[1] reflecting the model's uncertainty about what the

[1] For a normal distribution, entropy $h = \log \sigma + h_0$.

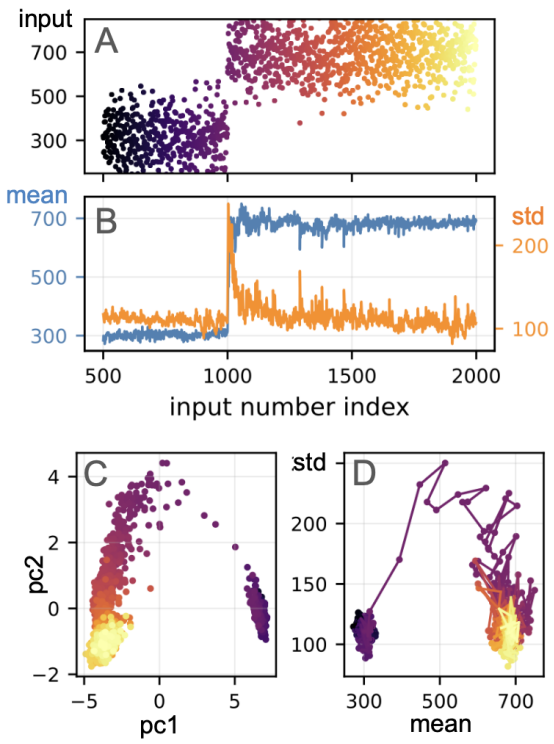

Figure 2: **Belief dynamics.** Input (A) is a time series with a sharp transition between two distinct regimes, $\mathcal{N}_{300,\sigma_0} \rightarrow \mathcal{N}_{700,\sigma_0}$ at $t = 1000$. Output (B) shows the model quickly adapting its mean after the switch while broadening its variance, which relaxes back to its true value after about 300 tokens. This switch in belief is apparent in the model's activations (C). The trajectory in probability space (D) shows two attractors corresponding to the true input distributions, and the path taken by the model across them.

input data represents.

In Appendix B, we compare the model's empirical response to that of an ideal observer. In Appendix C, we consider the response to a time series with multiple switches and observe meta-in-context learning capabilities.

3.3 Features encoded along data manifolds

With belief states mapped as manifolds and updates measured as trajectories, we investigate which coordinates of belief are accessible to simple readouts. This matters for two reasons: (i) probing: how we extract what the model "believes" without relying on prompting; and (ii) intervention: – how we define a principled steering objective, such as: "increase μ while preserving σ". We therefore study how the μ-parameter is decodable along the manifold using linear field probes.

3.3.1 Linear field probes

To address the mounting evidence that neural networks, and transformers in particular, encode features using geometries and topologies far more complex than single directions, Yocum et al. (2025) recently introduced *feature fields* to describe features defined over (non-Euclidean) manifolds. The framework proposes a natural extension of linear probes into families of probes, called *linear field probes* (LFP), which essentially operate a piecewise-linear tiling of an underlying manifold.

Using our continuous representation of beliefs, we examine the linear field probing on the gaussian-preimage manifolds $\mathcal{M}^{(l)}_{\mu,\sigma}$. We find that linear field probes produce an excellent representation of the belief manifolds.

Specifically, we train a set of linear probes for activations representing the discretized manifold $\sigma_0 = 100, \mu_i = \{300, 350, \ldots, 700\}$, at all layers (see Appendix D.2 for details). We report the following findings:

- **Separability**: probes achieve high accuracy on all activation classes μ_i, from 0.87 at layer 0 to 0.99 at layer 15 (Fig. 3A); this shows that μ-indexed representations are linearly separable at all layers.

- **Continuity**: probes vary smoothly with the parameter μ, as revealed by the structured cosine similarity matrix in Fig. 3B; this smooth similarity structure reflects an underlying geometry over the domain μ, as further discussed below.

- **Interpolation**: probe vectors can be interpolated across μ based on trained endpoints (Fig. 3C); for example, the probe at $\mu = 350$ can be interpolated between $\boldsymbol{w}_{300}$ and $\boldsymbol{w}_{400}$ without retraining. This indicates that the probes form a coherent field over μ rather than a collection of unrelated classifiers.

- **Transfer**: probes transfer *only locally* along the manifold; transfer performance decays with distance in μ at a rate matching the decay lengthscale in the Gram matrix (Fig. 3D). This indicates curvature in the domain embedding and limits the validity of a single (global) linear direction.

Together, these observations support the interpretation that the set of linear probes constitute a

LFP as defined in Yocum et al. (2025). Separability establishes that μ is linearly decodable, while continuity and interpolation establish that μ is linearly represented as a field. Thus, the manifold of gaussian activations is a feature field that is linearly represented.

We stress that, in general, a standard linear probe for (non-parametric) classes, for example {cats, dogs, horses, raccoons} would *not* show the continuous and interpolative structure that characterizes linear field probes. A LFP represents a bilinear form $f(\boldsymbol{x}, \mu) = \langle \boldsymbol{x}, \Psi(\mu) \rangle$, where $\Psi(\mu)$ denotes the probe vector associated with field value $\mu \in \mathcal{Z}$; evaluation of the field is simply an inner product $\langle \cdot, \cdot \rangle$.

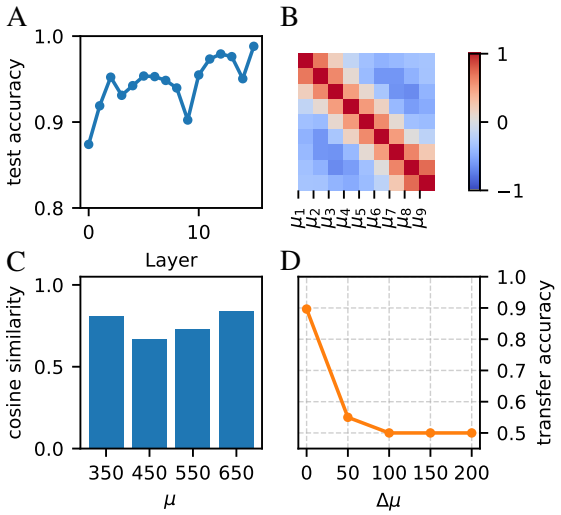

Figure 3: **Linear field probes.** (A) Separability: probe accuracy (on test set) on all μ-indexed representations, for each layer. (B) Continuity: cosine similarity between probes, showing smooth variation over μ and revealing structured geometry over the domain. (C) Interpolation: cosine similarity between true and (kernel-) interpolated (see D.3) vectors at intermediate μ; for reference, cosine similarity between two random vectors in a space of $d = 2048$ dimensions has mean 0 and standard deviation $1/\sqrt{2048} \simeq 0.02$. (D) Transfer: probes only transfer locally; here showing a probe trained on $\mu = \{300, 350\}$ (layer 0) and applied on sets $\mu = \{350, 400\} (\Delta\mu = 50)$, $\mu = \{400, 450\} (\Delta\mu = 100)$, etc. (An untrained binary probe has accuracy 0.5.)

3.3.2 Field geometry

In return, the linear field probe implicitly endows the domain $\mathcal{Z} = [300, 700]$ with a particular geometry. Importantly, this geometry does not describe where activations corresponding to different domain values are (co-)located in activation space. Instead, it describes which linear directions in activation space matter for decision boundaries: the way μ is encoded is not the same as the way μ can be read out. This distinction between encoding geometry (primal) and separability geometry (dual) is central to Yocum et al. (2025). We provide additional intuition and discussion of this subtle point in Appendix D.1.

The geometry of this linear readout space is captured by the Gram matrix of probe vectors (cosine similarity), shown in Fig. 3B. Its eigendecomposition reflects the underlying dimensionality and directionality necessary to decode μ values from the embedded representations. Following standard kernel PCA, this dual geometry can be visualized by embedding each μ_i as

$$\left(\sqrt{\lambda_1} \boldsymbol{u}_1(\mu_i), \sqrt{\lambda_2} \boldsymbol{u}_2(\mu_i), \sqrt{\lambda_3} \boldsymbol{u}_3(\mu_i) \right),$$

where λ_β and $\boldsymbol{u}_\beta$ are the leading eigenvalues and eigenvectors of the Gram matrix (Fig 4B).

Figure 4 thus illustrates a representation in which the readout geometry is dominated by a small number of principal directions, even when the underlying activation manifold is highly curved. This highlights the duality between encoding complexity and linear usability: complex internal representations can give rise to comparatively simple linear fields.

Finally, the LFP also reveals the evolution of the field geometry across layers. In particular, eigenvalues of the Gram matrix show that the intrinsic dimensionality of the manifold increases over layers (Fig. 4A) – with the exception of the last layer[2]. It may be interpreted as the deeper representations encoding an increasing number of features, requiring the feature field to densify and spread across more dimensions in readout space.

3.4 Interventions

Based on the model's belief map, can we purposefully intervene on its activations and engineer its output to match a specific distribution? This is traditionally referred to as *model steering*.

Model steering is usually performed along a specific direction with a tunable magnitude; in other words: *linearly*. Some have recently argued that steering ought to be done along the manifold that supports the underlying data rather than a constant direction (Hindupur et al., 2025). Alternatively, the

[2]This "last layer anomaly" (Sarfati et al., 2025) has been noted many times before, but remains to this day to be characterized and explained precisely. Intuitively, the model converges to a next token prediction.

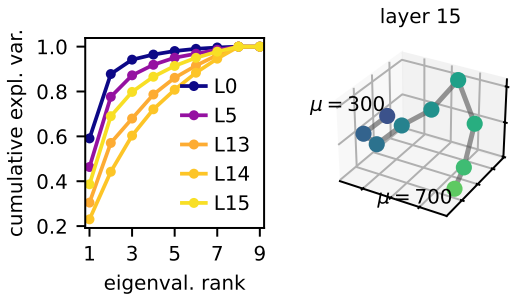

Figure 4: **Field geometry.** (Left) Cumulative variance explained as a function of rank of the eigenvalues of the LFP Gram matrices (Fig 3B), for each layer. The intrinsic dimensionality increases with layers, but drops at the last layer (L15). (Right) Kernel PCA embedding of the first 3 eigenvectors of the Gram matrix at layer 15. This represents the field geometry, dual to the activation space manifold.

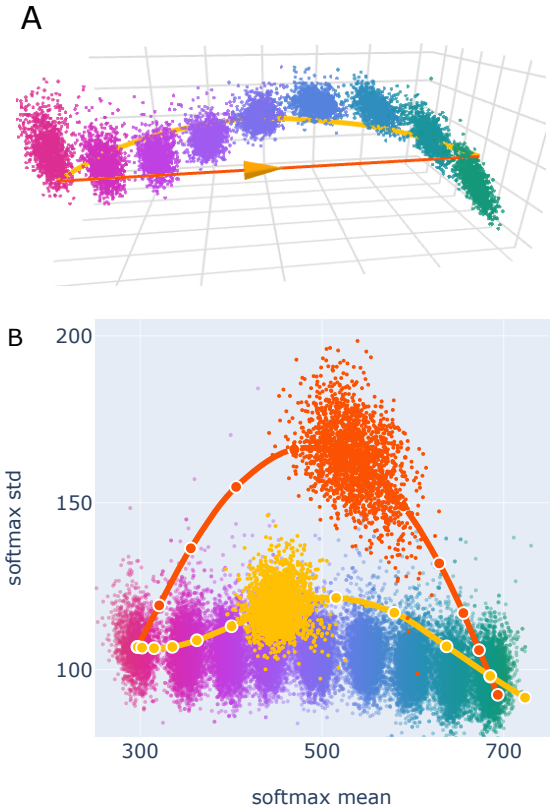

Figure 5: **Steering based on activation geometry: linear vs manifold-aware.** The manifold of activations (A) for $\mu \in [300, 700]$ provides a principled basis for steering directions, for example a centroid-to-centroid vector (orange), and a manifold-fitting spline (yellow). The corresponding resulting logits are shown in (B). Steering linearly brings steered logits far out of the $\sigma = 100$-manifold (orange path), while steering along the manifold keeps outputs significantly more aligned with the targeted behavior.

field geometry uncovered above suggests the possibility for *field-aware* steering relying on probe vectors.

Here, we use our experiments to compare various steering schemes based on the underlying geometry of the manifolds. We find that steering linearly tends to push next-token prediction out of the target distribution.

3.4.1 Activation-aware steering (primal space)

Assume we have an ensemble of last-layer representations $\boldsymbol{x}_{i,k}^{(300)} \in \mathcal{M}_{300,100}$ mapping onto next-token predictions $\sim \mathcal{N}_{300,100}$, and a second set $\boldsymbol{x}_{i,k}^{(700)} \in \mathcal{M}_{700,100}$ mapping onto $\mathcal{N}_{700,100}$. Suppose we aim to steer the $\mu = 300$ representations towards an intermediate target $\mathcal{N}(500, 100)$. A simple steering approach computes the "difference of means" steering vector

$$\boldsymbol{s} = \overline{\boldsymbol{x}}_{700} - \overline{\boldsymbol{x}}_{300},$$

where $\overline{\boldsymbol{x}}_{300}, \overline{\boldsymbol{x}}_{700}$ are class centroids, and applies the linear intervention::

$$\tilde{\boldsymbol{x}}_{i,k}^{(500)} \simeq \boldsymbol{x}_{i,k}^{(300)} + \alpha \boldsymbol{s},$$

with $\alpha = 0.5$ (halfway between 300 and 700). This is essentially Contrastive Activation Steering (Rimsky et al., 2024).

However, the set $\mathcal{M}_{\mu,100}$ of the preimages to $\mathcal{N}_{\mu,100}$ vary nonlinearly with μ, and the resulting trajectory is curved in activation space (Fig 5A). Thus, steering linearly towards $\mathcal{M}_{700,100}$ might push activations out of the gaussian manifold, so that when they reach $\mu = 500$ their shape departs significantly from that of the normal distribution $\mathcal{N}_{500,100}$. This is indeed what we observe experimentally in Fig. 5B, where the linear steering in activation space forms a curve in logit (image) space so that steered activations reaching $\mu = 500$ have a large variance (orange curve).

In contrast, if we parametrize along the prototypes of $\mathcal{M}_{\mu,\sigma_0=100}$ and steer *along the manifold*, we preserve the class-conditioned structure: induced predictions remain close to $\mathcal{N}(\mu, \sigma_0)$ while shifting their mean towards $\mu = 500$ (Fig. 5B yellow curve).

This is concrete evidence that linear steering can lead to unexpected and unpredictable behaviors when the underlying geometry is not well-characterized.

3.4.2 Field-aware steering (dual space)

A complementary approach to steering acts in the *dual* (feature) space using probe vectors. Rather than steering towards prototypes in activation space, this method attempts to modify the component of the representation that controls a linear readout of μ.

Linear probe steering. In our setup, we can use point probes to define the steering vector:

$$\boldsymbol{s}^\star = \frac{\boldsymbol{w}_{\mu=700} - \boldsymbol{w}_{\mu=300}}{\|\boldsymbol{w}_{\mu=700} - \boldsymbol{w}_{\mu=300}\|}.$$

Steering $\boldsymbol{x}^{(300)} \in \mathcal{M}_{300}$ towards $\mathcal{M}_{700}$ reads:

$$\boldsymbol{x}_s(\alpha) = \boldsymbol{x}^{(300)} + \alpha \boldsymbol{s}^\star.$$

One immediate caveat is that, since *feature geometry is agnostic to scale*, it's unclear which values of α are meaningful. Prior work has found that small α can have negligible effects on logits (StefanHex and Mendel (2024)'s "activation plateaus"), whereas large values break the model towards distribution shifts and degenerate outputs.

We directly visualize the evolution of the induced output distribution as a function of α in Fig. 4. Again, this naive linear steering scheme pushes output distributions away from the Gaussian manifold.

Field-aware steering from LFP. Alternatively, the linear field probe provides a geometry over μ to extract field-aware steering vectors. Specific implementations vary; we propose the following. We use the first r eigenmodes of the LFP Gram matrix $\boldsymbol{K}$ and obtain a low-dimensional embedding:

$$c(\mu) = \left(\sqrt{\lambda_1}\boldsymbol{u}_1(\mu), \ldots, \sqrt{\lambda_r}\boldsymbol{u}_r(\mu)\right).$$

We then fit a spline $\tilde{\boldsymbol{c}}(\mu)$ and use it to define an interpolated steering direction $\boldsymbol{s}^\star = \sum_i \alpha(\mu)\boldsymbol{w}_{\mu_i}$, where the weights α_i are obtain by kernel regression using $\boldsymbol{K}$. This removes noisy components and preserves only the smooth and predictive geometry.

As shown in Fig. 4, this approach maintains the induced output distribution closer to the intended $\mathcal{N}_{\mu,\sigma_0}$ family, within a certain range.

Field-constrained interventions. Another promising approach would intervene only on the component of activations that lives in probe subspace by defining a projector onto the probe span. The field-constrained steering would hence preserve all orthogonal information that does not align with the steering objective. This is out-of-scope for this current work.

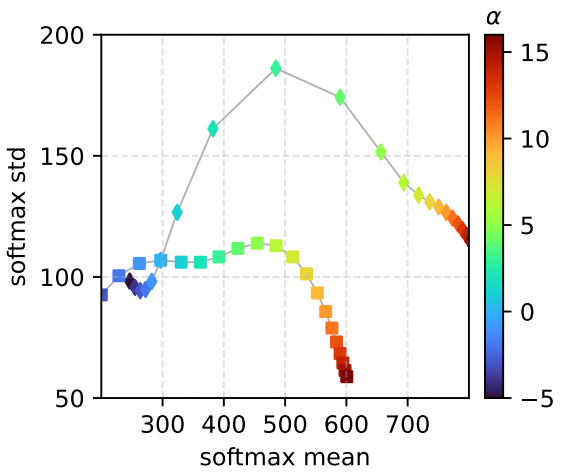

Figure 6: **Steering based on feature and field geometry.** Steering along a single probe direction, here $\boldsymbol{w}_{300\to700}$ (diamonds) again brings the steered activations far from the $\sigma_0 = 100$ manifold. Taking advantage of the linear field probe to smoothly travel along the field geometry (squares), however, maintain the same standard deviation while increasing the mean – at least in a certain regime between 300 and 500.

4 Discussion

The Linear Representation Hypothesis (LRH) is a convenient framework to think of latent representations as a linear sum of features encoded as directions. However, ample evidence has found that features are often encoded along curved manifolds, such as circular supports (Gurnee et al., 2025). Our previous observations herein concur.

Hence the question: given manifold parametrizations, can we instead interpolate any combination of parameters as a *mixture of manifolds*? More precisely, around an anchor (μ_0, σ_0), we hypothesize that the effect of μ and σ on the representations decomposes into independent subspaces U and V such that

$$c(\mu, \sigma) \simeq c(\mu_0, \sigma_0) + u(\mu) + v(\sigma),$$

with $u(\mu) \in U, v(\sigma) \in V$, and the gauge fixed at $u(\mu_0) = v(\sigma_0) = 0$. Under this model, the surface of prototypes is approximately a product manifold embedded as an additive superposition. We investigate experimentally by fitting a spline $c_\mu(\mu)$ through the centroids of the $\mathcal{M}_{\mu,100}$ tiles and another spline $c_\sigma(\sigma)$ across $\mathcal{M}_{500,\sigma}$. Then, we

interpolate:

$$\tilde{c}(\mu^*, \sigma^*) = c_0 + (c_\mu(\mu^*) - c_\mu(\mu_0)) + (c_\sigma(\sigma^*) - c_\sigma(\sigma_0)),$$

with $c_0 = c(\mu_0, \sigma_0)$ denoting an anchor point, for example $(\mu = 500, \sigma = 100)$. Fig. 7 shows the interpolated centroids compared to the ground-truth centroids computed from the activations of the corresponding time series. Evidently, the simple, spline-based interpolation fails to capture the true geometry of the (μ, σ) sheet. This is possibly due to non-linear interactions between parameters, here μ and σ.

Other decompositions might be possible. For example, Fel et al. (2025) recently proposed the Minkowski Representation Hypothesis, according to which concepts are represented as convex hulls, and representations can de decomposed as (non-unique) sums of polytopes.

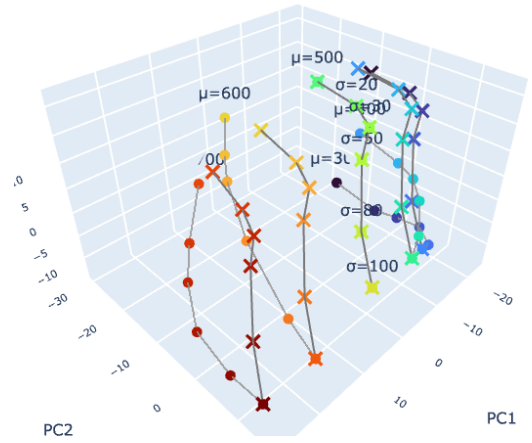

Figure 7: **Mixture of manifolds.** Prototypes interpolated from the anchor manifolds $\mathcal{M}_{\mu,\sigma_0}$ and $\mathcal{M}_{\mu_0,\sigma}$ with $(\mu_0 = 500, \sigma_0 = 100)$, marked with a cross, do not match the geometry of the ground-truth centroids (circles).

5 Conclusion

We introduced a principled framework for tracking language models' beliefs by mapping how posterior beliefs are encoded as structured manifolds in activation space, and how non-stationary inputs induce dynamics between attractor regions. Using this setup, we showed the linear field probes naturally capture belief manifolds and that their induced field geometry provides guidance for inference-time interventions.

Our main finding is that purely linear concept representations are often an inadequate abstraction. Even when a target feature, such as the mean of an output distribution, is locally decodable, the underlying representations can exhibit substantial curvature. Hence, linear steering can move activations off-manifold, producing unintended coupled shifts in other covariates, such as the variance or shape of the distribution. This geometric mismatch offers a concrete explanation for why steering methods can be brittle and why controlling a certain behavior can affect others.

Our analysis relies on true activations of base models without finetuning; yet, it is currently constrained to numerical settings. Extending these findings to broader natural language processing (NLP) contexts remains open, particularly because identifying continuous parametrizations in natural language is more challenging.

Accordingly, our results should be interpreted as clarifying what internal representations look like geometrically, and what geometry implies for probing and steering, rather than as a complete account of belief formation in general-purpose NLP.

Limitations

We rely on small open-source base models and focus on a restricted family of prompt distributions: time-series. While this improves interpretability, it may not capture the diversity of internal geometries induced by other model families, tokenization schemes, or real-world prompting regimes. The core of our methodology relies on the principle that beliefs live on manifolds that can be indexed by interpretable parameters, such as μ and σ. Linear field probes and the induced field geometry are particularly well-suited to such settings, but it remains unclear how broadly this assumption holds in natural language tasks, where the relevant latent may not be fully parameterizable. Finally, in this paper, we do not investigate causal attribution to uncover which neuronal structure or circuits are responsible for the geometry observed.

References

Jiajun Bao, Nicolas Boullé, Toni J. B. Liu, Raphaël Sarfati, and Christopher J. Earls. 2025. Text-trained llms can zero-shot extrapolate pde dynamics, revealing a three-stage in-context learning mechanism. *Preprint*, arXiv:2509.06322.

Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and

Tomer Ullman. 2024. Forking paths in neural text generation. *Preprint*, arXiv:2412.07961.

Eric Bigelow, Daniel Wurgaft, YingQiao Wang, Noah Goodman, Tomer Ullman, Hidenori Tanaka, and Ekdeep Singh Lubana. 2025. Belief dynamics reveal the dual nature of in-context learning and activation steering. *Preprint*, arXiv:2511.00617.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matthew Botvinick, Jane X. Wang, and Eric Schulz. 2023. Meta-in-context learning in large language models. *Preprint*, arXiv:2305.12907.

Thomas Fel, Binxu Wang, Michael A. Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep S. Lubana, Talia Konkle, Demba Ba, and Martin Wattenberg. 2025. Into the rabbit hull: From task-relevant concepts in dino to minkowski geometry. *Preprint*, arXiv:2510.08638.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. 2025. When models manipulate manifolds: The geometry of a counting task. *Transformer Circuits Thread*.

Peter Gärdenfors. 2000. *Conceptual Spaces: The Geometry of Thought*. The MIT Press.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba E. Ba. 2025. Projecting assumptions: The duality between sparse autoencoders and concept geometry. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale*.

Junjie Hu, Gang Tu, ShengYu Cheng, Jinxin Li, Jinting Wang, Rui Chen, Zhilong Zhou, and Dongbo Shan. 2025. Harp: Hallucination detection via reasoning subspace projection. *Preprint*, arXiv:2509.11536.

Toni J.b. Liu, Nicolas Boulle, Raphaël Sarfati, and Christopher Earls. 2024. LLMs learn governing principles of dynamical systems, revealing an in-context neural scaling law. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15097–15117, Miami, Florida, USA. Association for Computational Linguistics.

Toni J.B. Liu, Nicolas Boulle, Raphaël Sarfati, and Christopher Earls. 2025. Density estimation with LLMs: a geometric investigation of in-context learning trajectories. In *The Thirteenth International Conference on Learning Representations*.

Julian Minder, Kevin Du, Niklas Stoehr, Giovanni Monea, Chris Wendler, Robert West, and Ryan Cotterell. 2025. Controllable context sensitivity and the knob behind it. *Preprint*, arXiv:2411.07404.

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

Katherine N. Quinn, Colin B. Clement, Francesco De Bernardis, Michael D. Niemack, and James P. Sethna. 2019. Visualizing probabilistic models and data with intensive principal component analysis. *Proceedings of the National Academy of Sciences*, 116(28):13762–13767.

Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. 2025. LLM-microscope: Uncovering the hidden role of punctuation in context memory of transformers. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7757–7764, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Raphaël Sarfati, Toni J.B. Liu, Nicolas Boulle, and Christopher Earls. 2025. Lines of thought in large language models. In *The Thirteenth International Conference on Learning Representations*.

StefanHex and Jake Mendel. 2024. Interim research report: Activation plateaus & sensitive directions in gpt2. LessWrong blog post.

Julian Yocum, Cameron Allen, Bruno Olshausen, and Stuart Russell. 2025. Neural manifold geometry encodes feature fields. In *NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations*.

A Convergence to the input distribution

A prompt consisting of n numbers contains $2n + 1$ tokens: an initial `<|begin_of_text|>` token, n *com2num* tokens, and n *num2com* tokens. We denote by $t \geq 0$ the index of each input number,

i.e. each number token has global index $2t+1$ (0-indexed). The corresponding *com2num* tokens predicting the following numbers have index $2t+2$.

Figure 8 shows the convergence of the `com2num` outputs $\boldsymbol{p}_t = \boldsymbol{p}(2t+2)$ towards the true distribution underlying the data. After about 100 numbers have been seen, the model adequately reproduces the underlying distribution in its logits. Figure 9 further shows that outputs are locally close to each other, suggesting continuity of the output softmax along t.

Liu et al. (2025) and Bao et al. (2025) have already studied the mechanisms of convergence in various numerical settings, starting with syntactic matching and a high-entropy phase corresponding to the uniform distribution. We further note here that even 100 input numbers is insufficient to sketch an underlying distribution with $\sigma = 100$, even though the model's output is faithful to it. In other words, the model *believes* the input data is Gaussian, even though it doesn't yet look like it. This suggests the model's *prior* is biased towards normality.

B Ideal observer

In a previous study, Liu et al. (2025) observed that LLMs operate an implicit, context-dependent Kernel Density Estimation procedure to compute logit distributions from a numerical input. Here, we consider the perspective of a Bayesian observer which incorporates the sequentiality of input data to infer its own posterior.

Considering the switching dynamics of Section 3.2, it is apparent that the language model does *not* conform to an online ideal Bayesian observer under standard assumptions. For one thing, intuitively, and ideal observer would compute a running average of the input data and hence the output distribution mean would not converge fast enough to 700 as in Fig 2.

More precisely, assuming the model's generative model is stationary & independent Gaussian, with unknown mean and unknown variance, the output distribution should follow:

$$\sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right) \mid \sigma^2 \sim \text{InvGamma}(\alpha_0, \beta_0).$$

The resulting posterior $p(x_{t+1}|x_{1:t})$ is StudentT distributed, with the posterior mean converging as $1/t$ and the posterior standard deviation converging above 100 as it attempts to bridge a bimodal distribution with equal halves.

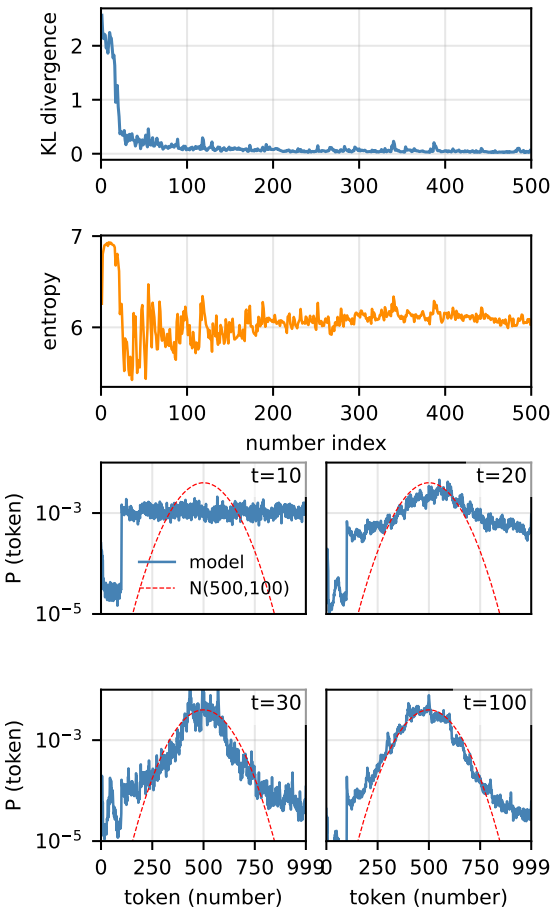

Figure 8: **Convergence to the normal distribution.** (A) KL divergence between the softmax distribution and the distribution underlying the input numbers, $\text{KL}(\boldsymbol{p}_t \,\|\, \mathcal{N}_{500,100})$. The convergence appears to be reached after about 100 numbers. (B) Entropy of the softmax distribution. (C) Output distributions at different times, showing convergence from a uniform distribution on integers to the true distribution (red).

Alternatively, a Kalman filter with constant $\sigma = 100$ would result in a sharp convergence of the mean, but would lack any variance shift.

C Meta-in-context learning

In Fig. 2, we observed the response of the LLM upon a change of input distribution. We now extend this experiments to a long sequence of several switches between $\mathcal{N}(300, 100)$ and $\mathcal{N}(700, 300)$. The main question here is whether the model starts to understand the *meta-distribution* of the input data. Figure 10 provides early evidence that it does. Indeed, the model's response to a change of distribution becomes faster and faster, as evidenced in the shape of the softmax standard deviation over time, and the trajectories of activations between the

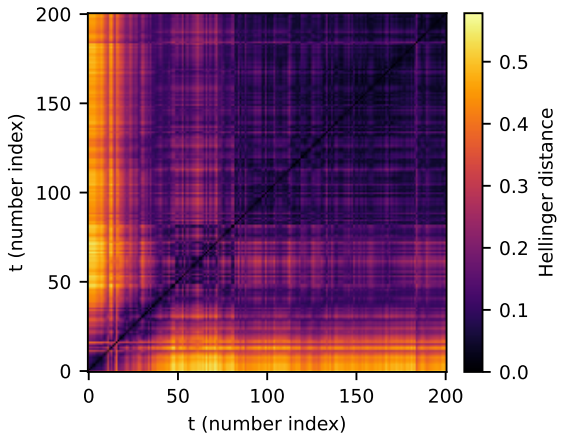

Figure 9: **Pairwise distances between output distributions.** The Hellinger distance is a (symmetric) distance between two probability vectors: $D_{\text{Hel}}(p_t, p_{t'}) = 2^{-1/2}\|\sqrt{\boldsymbol{p}_t} - \sqrt{\boldsymbol{p}_{t'}}\|_2$. Early outputs are far from those at $t \geq 100$, which encode the stationary distribution.

two attractors. The model transitions much faster and follows a more direct path for later switches and the early ones. This ability of *meta-in-context learning* has been reported before, in different settings (Coda-Forno et al., 2023).

D Linear field probes

D.1 Data and feature geometry

In order to illustrate the dichotomy between data manifold and field geometry (primal vs dual), as framed in Yocum et al. (2025), we propose the visualization in Fig. 11. Suppose representations lie on a helicoid: their geometry is fully three-dimensional. Let's now assume that these activations describe separate classes: blue, orange, green; class assignment varies along the manifold There are many ways these classes can be encoded along the manifold. In Fig 11A, B, D, E these classes are *linearly separable* because there exists linear probe vectors (equivalently: single hyperplanes) that separate the classes. In Fig 11C, no such linear separator exists, therefore the classes are not separable from single decision boundary directions.

Let's now consider the dimensionality of the linear field probe. In Fig 11D, the class probe vectors are collinear across μ, so that $\dim \text{span}\{w_\mu\} = 1$ (a rank-1 field). However, in Fig 11E, two *non-parallel* hyperplanes are required, one to separate blue from orange, and another for orange to green: the span of the linear field probe is 2. This intrinsic field dimensionality is reflected in the rank of the LFP Gram matrix.

D.2 Set of linear probes

Practically, to calculate a set of linear probes on an ensemble of activations representing a set of C classes, and test the hypothesis that they constitute a LFP, we have two main options:

A multiclass probe, which computes the logits:

$$\boldsymbol{z}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b},$$

with $\boldsymbol{z} \in \mathbb{R}^C, \boldsymbol{W} \in \mathbb{R}^{C\times d}, \boldsymbol{b} \in \mathbb{R}^C$. It is trained using a softmax + cross-entropy loss. The point probes correspond to the domain embedding vectors $\Psi(\mu_i)$ at the discretized points μ_i, i.e. $\Psi(\mu_i)$ are the rows $\boldsymbol{w}_i$ of $\boldsymbol{W}$.

A one-vs-rest probe set, i.e. a set of C binary classifiers with scores for each class i:

$$s_i(\boldsymbol{x}) = \boldsymbol{w}_i^\top \boldsymbol{x} + b_i,$$

and corresponding probabilities $p_i(\boldsymbol{x}) = \sigma\left(s_i(\boldsymbol{x})\right)$ and a sigmoid + binary cross-entropy loss. The point probes are the C learned $\boldsymbol{w}_i \in \mathbb{R}^d$.

In either case, the bias term b is optional and a weight-decay is applied for training. We set $\boldsymbol{b} = 0$ for our probes for simplicity – it doesn't seem to affect performance. In practice, we find one-vs-rest probes to be less efficient to separate a specific class from all others. Hence we rely on the multiclass probe in what follows.

The steering directions are calculated from differences of point-probes. (For multiclass probes using a softmax model, the row vectors are defined up to a shared shift, so we optionally center them across classes without affecting steering differences.)

D.3 Manifold interpolation

We consider the unit-norm probe vectors $\boldsymbol{w}_{\mu_i}$; for simplicity let's take $\boldsymbol{w}_{300}$ and $\boldsymbol{w}_{400}$. We aim to find an interpolation for the probe vector $\tilde{\boldsymbol{w}}_{350}$ at $\mu = 350$.

Linear interpolation. This is simply the arithmetic weighted mean: $\tilde{\boldsymbol{w}}_{350} = \alpha\boldsymbol{w}_{300} + (1-\alpha)\boldsymbol{w}_{400}$, with $\alpha = 0.5$ for the midpoint.

Geodesic interpolation. We use the standard spherical interpolation formula:

$$\tilde{\boldsymbol{w}}_{350} = \frac{\sin\left(1-\alpha\right)\theta}{\sin\theta}\boldsymbol{w}_{300} + \frac{\sin\alpha\theta}{\sin\theta}\boldsymbol{w}_{400},$$

where $\theta = \arccos \boldsymbol{w}_{300}^\top \boldsymbol{w}_{400}$ and $\alpha = 0.5$.

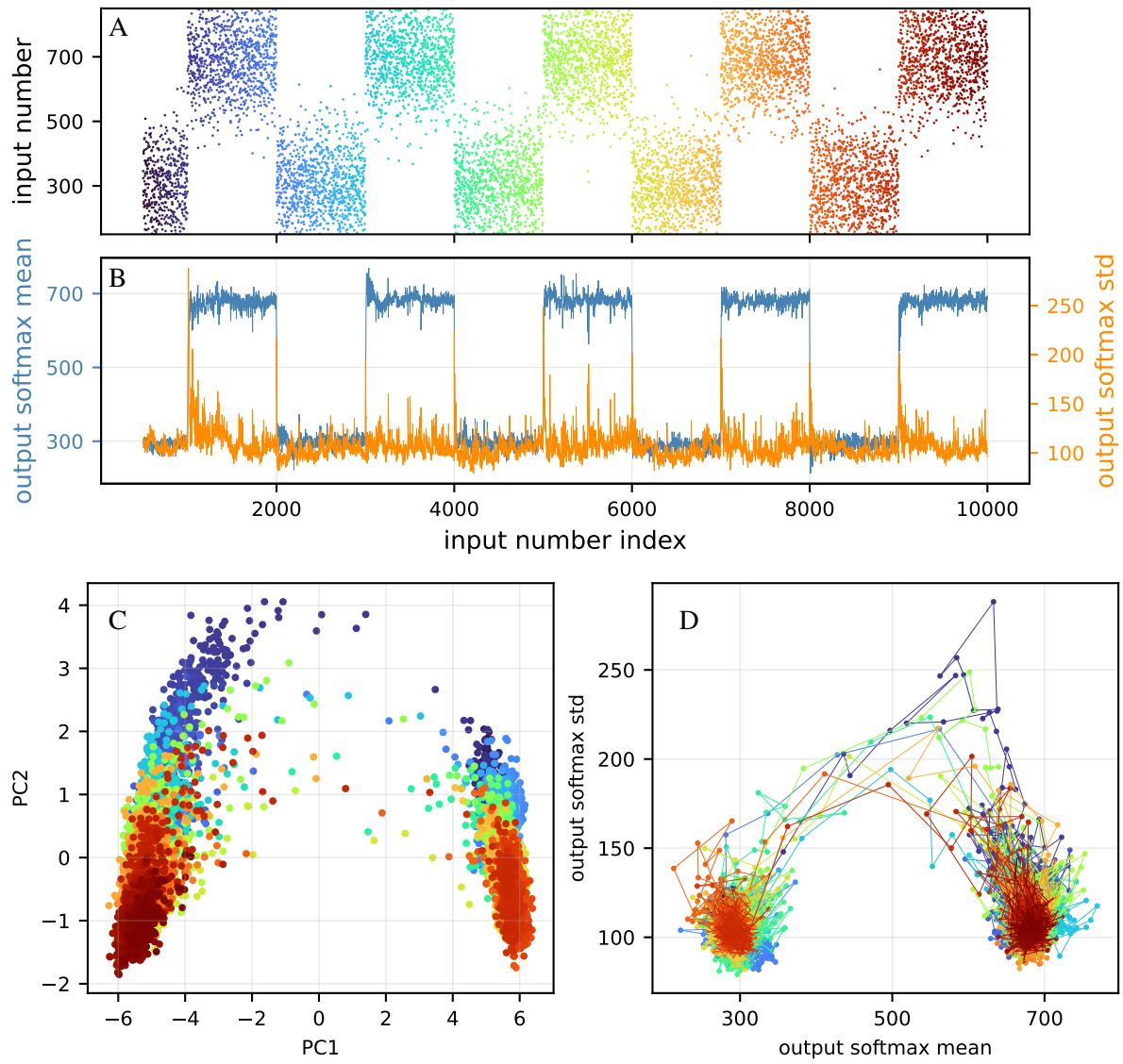

Figure 10: **Meta-in-context learning.** (A) The input time-series consists of 10 sequences of 1000 numbers each, alternating between $\mathcal{N}(300, 100)$ and $\mathcal{N}(700, 100)$. (B) Mean and standard deviation of the softmax output (next-number prediction). (C) Activations (layer 14) for the corresponding input tokens in (A). (D) Trajectory in the (mean,std) plane of the output distributions , colored by input index.

Kernel interpolation. In this case, $\tilde{\boldsymbol{w}}_{350} = \sum_i a_i \boldsymbol{w}_i$, with $\boldsymbol{a} = \boldsymbol{G}^{-1}\boldsymbol{k}_{350}$ and $\boldsymbol{k}_{350} = \alpha \boldsymbol{G}_{300,j} + (1-\alpha)\boldsymbol{G}_{400,j}$, where $\boldsymbol{G}$ is the Gram matrix (the notation $\boldsymbol{G}_{300,j}$ is shorthand for $\boldsymbol{G}_{(i,j)|\mu_i=300}$, i.e. the row of $\boldsymbol{G}$ corresponding to μ=300).

Eigenbasis interpolation. Yocum et al. (2025) propose a more rigorous approach from their Field Geometry Equivalence Theorem based on spectral decomposition in feature space.

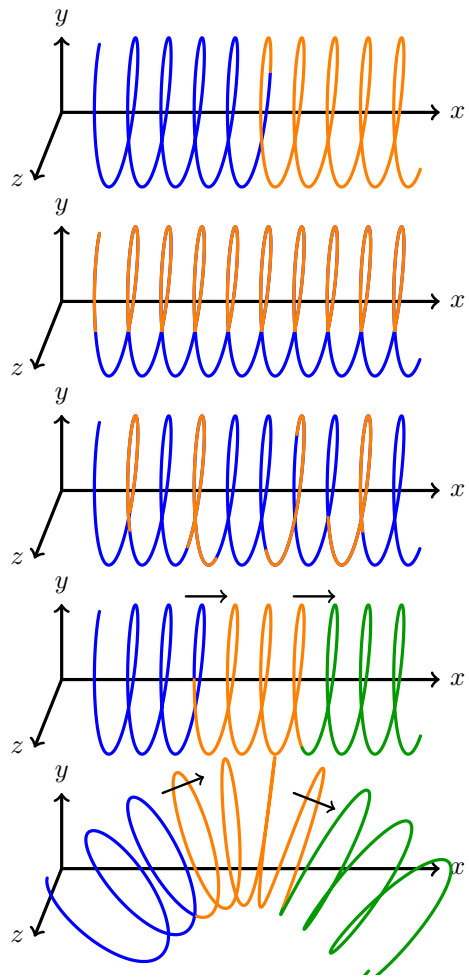

Figure 11: **Data geometry vs field geometry.** (A) The data forms a 3D helicoid, yet the two classes (blue/orange) depend only on the position along the x-axis: they are linearly separable (by a hyperplane perpendicular to the x-axis). Separability does not require the data manifold to be flat. (B) Another example of linearly separable classes, along the y-axis (xz-plane). (C) Non-linearly separable classes: no single hyperplane separates the two colors. (D) The three classes are linearly separable, with probe vectors being all collinear: the linear field probe is rank 1. (E) Curved field geometry: the probe directions rotate with μ, requiring at least a 2D probe subspace.