Neural Manifold Geometry Encodes Feature Fields

Julian Yocum JULIANY@BERKELEY.EDU
Cameron Allen CAMALLEN@BERKELEY.EDU
Bruno Olshausen BAOLSHAUSEN@BERKELEY.EDU
Stuart Russell RUSSELL@BERKELEY.EDU
UC Berkeley

Editors: List of editors' names

Abstract

Neural networks represent concepts, or "features", but the general nature of these representations remains poorly understood. Previous approaches treat features as scalar-valued random variables. However, recent evidence for emergent world models motivates investigating when and how neural networks represent more complex structures. In this work, we formalize and study *feature fields*—function-valued features defined over manifolds and other topological spaces corresponding to the underlying world (e.g., value functions, belief distributions). We introduce *linear field probing*, a method that extends linear probing to extract feature fields from neural activations. Whereas a linear probe maps scalar features to individual points in activation space, a linear field probe embeds the topological space of a feature field into activation space. We prove that the geometry of this embedding fully defines the space of linearly representable functions for a given feature field. We empirically study feature fields of various topologies using linear field probing and present evidence of their emergence in transformers. This work establishes a formal connection between geometry and representation in neural networks.

Keywords: Neural Representations, Interpretability, Linear Probes, World Models

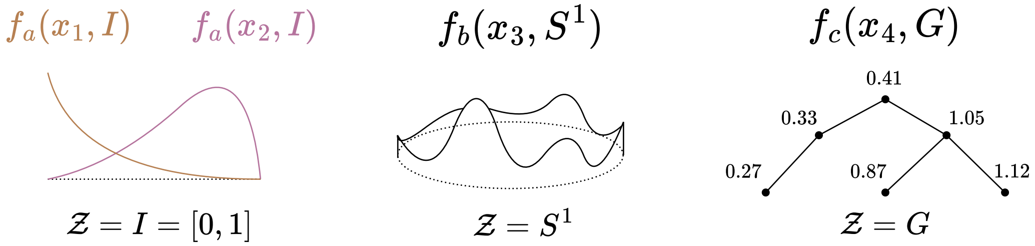

Figure 1: An illustration of three feature fields labeled a, b, c on distinct domain spaces: the interval I, the circle S^1, and the graph G. For every data point $x \sim X$, the field is realized as a function over the domain space: $z \mapsto f(x, z) \in \mathbb{R}^{\mathcal{Z}}$. Feature field a depicts two realizations for each of x_1, x_2. Feature fields may represent posterior distributions over a parameter space, or value functions over a state space.

1. Introduction

Within AI and neuroscience, *world models* or *cognitive maps*, respectively, are representations that reflect structural relationships in the world. Recent work shows that these world models emerge in neural networks during learning Gurnee and Tegmark (2023); Nanda et al. (2023). For example, Li et al. (2023) discovered a board game world model in a network trained to blindly predict sequences of game moves, despite never seeing the board or being told the rules of the game. Much effort has been made on uncovering the hypothesized atoms or *features* of neural representation. In activation space, these features appear to have highly structured geometry (Engels et al., 2025; Li et al., 2025; Park et al., 2024a; Shai et al., 2024; Li et al., 2023). However, many approaches typically assuming these features to be independent or *separable* (Elhage et al., 2022; Park et al., 2024b).

To investigate the relational structure such geometry may encode, we introduce *feature fields*, a representation whose atoms are intrinsically related by an underlying topological space, as illustrated in Figure 1. We make the following contributions. First, we prove that continuous feature fields are represented as topological embeddings of their domain into activation space (Section 3.1). Second, we prove that the geometry of this embedding fully determines the space of representable functions: feature fields live in a finite-dimensional reproducing kernel Hilbert space whose basis functions are encoded by the domain embedding geometry (Section 4). Third, we introduce *linear field probing*, a method for extracting feature fields from neural activations by discovering domain embeddings (Section 3.2). Finally, we empirically validate our framework by tracing the emergence and geometric evolution of domain embeddings across layers and training in transformers (Section 5).

2. Feature Variables

Consider the brain of a hypothetical dog. Her raw visual data about an object (such as a bone) consists only of signals from retinal neurons, yet the neural activity in her visual cortex may encode various meaningful *features* she perceives, such as "object weight (kg)" or "object length (m)." For these *scalar-valued* features, we define a map from data points in *data space* $x \in \mathcal{X}$ (e.g., retinal stimulus) to their corresponding scalar value (e.g., 1.03 kg), which we formalize as follows.

Definition 1 *Let $X : \Omega \to \mathcal{X}$ be a random variable called the* data distribution *associated with a* data space $\mathcal{X}$. *A* ***feature variable*** $F_i : \Omega \to \mathbb{R}$ *for a concept i is a scalar-valued random variable induced by a deterministic scalar map $f_i : \mathcal{X} \to \mathbb{R}$, with $x \mapsto f_i(x)$.*

Notice that this definition does not yet reference any neural network, as features may exist as a property of the data without being represented. We say a neural network *represents* a feature variable F_i if the scalar $f_i(x)$ can be recovered from the network's activation $\Phi(x)$ for any input $x \in \mathcal{X}$. These activations reside in a d-dimensional *activation space* $\mathcal{A} \subset \mathbb{R}^d$, where the network maps each input $x \in \mathcal{X}$ to an activation vector $\Phi(x) \in \mathcal{A}$. In the simplest case, the feature variable can be recovered via a dot product with the activations.

Definition 2 *A feature variable F_i is* ***linearly represented*** *in Φ if there exists a vector*[1] *$\Psi_i \in \mathbb{R}^d$ such that $f_i(x) = \langle \Phi(x), \Psi_i \rangle_{\mathcal{A}}$ for all $x \in \mathcal{X}$, where the inner product $\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\mathcal{A}} = \boldsymbol{v} \cdot \boldsymbol{w} + w_d$ is the standard Euclidean dot product augmented with a constant offset.*[2]

The implication is that **a feature variable is (linearly) represented by a vector** $\Psi_i \in \mathcal{A}$. This is operationalized by *linear probing* (Alain and Bengio, 2016; Belinkov, 2022), a method to identify this vector by training Ψ_i to minimize the approximation error $|f_i(x) - \langle \Phi(x), \Psi_i \rangle_{\mathcal{A}}|$ across the data distribution. A feature variable is linearly represented if and only if such a linear probe can successfully recover it from the network's activations. We illustrate two linearly represented feature variables in Figure 2 (top).

3. Feature Fields Are Topological Representations

Real-world data contains many features. As a result, prior work often represents them as collections of distinct feature variables. However, treating feature variables as independent can obscure important structural dependencies between them.

Suppose our dog is tracking the direction in which she left her bone. Rather than tracking only the single most likely direction, she may instead maintain a probability distribution f over the circular domain of possible compass directions, $z \in S^1$. As she acquires new data x, she may update f, so that, in particular, $f(x, z_{\mathrm{NE}})$ is her current probability that she left her bone toward the Northeast (see Figure 2). The topology of the domain (see Appendix A for a definition) may reflect essential structure of distributions f defined over it. For example, probabilities at neighboring directions may be continuous, i.e., $f(x, z_{\mathrm{NE}}) \approx f(x, z_{\mathrm{NNE}})$.

To capture such interdependencies, we introduce the *feature field.*

Definition 3 *A* ***feature field*** *$F_{\mathcal{Z}} : \Omega \to \mathbb{R}^{\mathcal{Z}}$ over a (topological)* ***domain space*** *$\mathcal{Z}$ is a random field (see Appendix A) induced by a deterministic scalar map*[3]

$$f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}, \quad (x, z) \mapsto f(x, z).$$

Two equivalent but insightful perspectives are obtained by fixing each of the arguments:
(i) (*Global view*) for every data point $x \sim X$, the feature field assigns a function over the domain $F_{\mathcal{Z}}(x) := (z \mapsto f(x, z)) \in \mathbb{R}^{\mathcal{Z}}$, called the **realization** of the feature field at x;
(ii) (*Local view*) for every domain point $z \in \mathcal{Z}$, the feature field induces a feature variable F_z via $(x \mapsto f(x, z))$. Thus, $F_{\mathcal{Z}}$ is a collection of feature variables indexed by points in $\mathcal{Z}$.

Consequently, a feature field may be understood either as a function-valued random variable with function domain $\mathcal{Z}$, or as a collection of feature variables indexed by points of that space, respectively. We illustrate three feature fields in Figure 1 and describe possible examples of their semantics as posterior distributions and value functions in Appendix B.

[1] Ψ_i may also be considered a *covector* from the activation dual space $\mathcal{A}^*$.

[2] Equivalently, we can augment the activation vector with a constant-valued dimension and incorporate the offset term into the weight vector.

[3] Throughout this paper, we use the term feature field to refer directly to the deterministic map f. The distinction between f and $F_{\mathcal{Z}}$ should be evident from context.

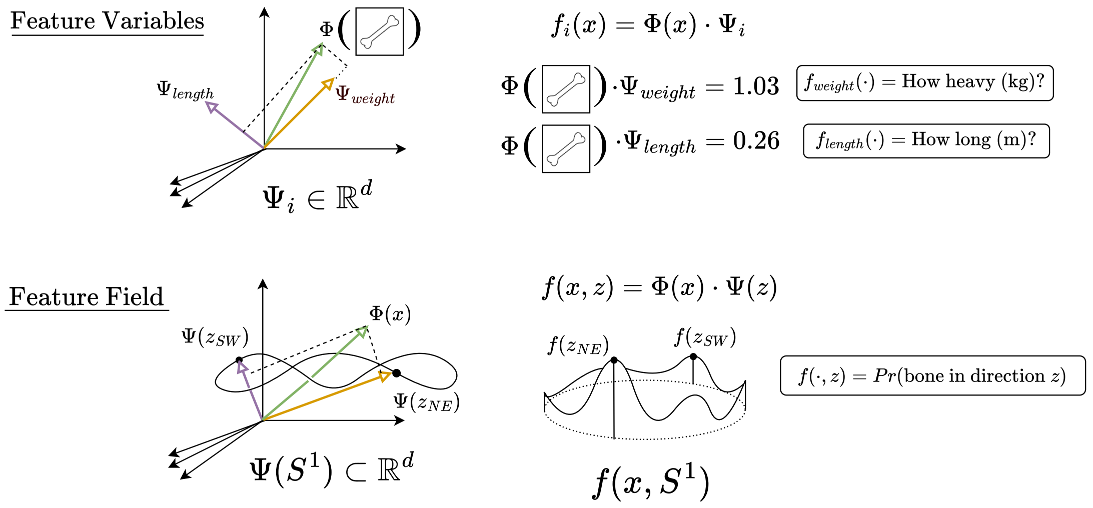
Feature Variables
Φ
Ψ_{length}
Ψ_{weight}
$\Psi_i \in \mathbb{R}^d$
$f_i(x) = \Phi(x) \cdot \Psi_i$
$\Phi(\ ) \cdot \Psi_{weight} = 1.03$
$f_{weight}(\cdot)$ = How heavy (kg)?
$\Phi(\ ) \cdot \Psi_{length} = 0.26$
$f_{length}(\cdot)$ = How long (m)?
Feature Field
$\Psi(z_{SW})$
$\Phi(x)$
$\Psi(z_{NE})$
$\Psi(S^1) \subset \mathbb{R}^d$
$f(x, z) = \Phi(x) \cdot \Psi(z)$
$f(z_{NE})$
$f(z_{SW})$
$f(\cdot, z) = Pr$(bone in direction z)
$f(x, S^1)$

Figure 2: An illustration of two feature variables and a feature field which are linearly represented within the activation space of the brain of a hypothetical dog. Top: A feature variable is represented by a vector Ψ_i that can read off a scalar function $f_i(x)$ from the activation $\Phi(x)$ by a dot product $f_i(x) = \Phi(x) \cdot \Psi_i$. For example, a dog's neural activations may encode information about the weight and length of an object she sees. Bottom: A *feature field* $f(x, z)$ is represented by a *domain embedding* $\Psi(\mathcal{Z})$ with $f(x, z) = \Phi(x) \cdot \Psi(z)$. For example, a dog tracking which way she left her bone may represent a whole distribution of probabilities over orientations $\mathcal{Z} \cong S^1$, where $f(x, z_{\text{NE}})$ is the probability she left it toward the Northeast given the history of her experience x.

3.1. The Domain Embedding

We have so far only defined a feature field as a property of the data itself, rather than as a representation in a neural network. In Section 2, we observed that a neural network may (linearly) represent a feature variable by a vector $\Psi_i \in \mathcal{A}$. From the local view, this definition naturally extends so that a feature field may have a representation where each point in the domain $z \in \mathcal{Z}$ is associated with a feature variable F_z and the vector $\Psi(z)$ representing it, resulting in a representation which is the collection of vectors $\{\Psi(z)\}_{z \in \mathcal{Z}}$.

Definition 4 *A feature field is* ***linearly represented*** *in* Φ *if and only if there exists a* ***domain map*** $\Psi : \mathcal{Z} \to \mathcal{A}$ *such that*

$$f(x, z) = \langle \Phi(x), \Psi(z) \rangle_A \quad \textit{for all } x \in \mathcal{X}, z \in \mathcal{Z}. \tag{1}$$

Equivalently, a feature field is linearly represented if and only if the feature variable F_z *at each point* $z \in \mathcal{Z}$ *is linearly represented.*

It follows that the image of the domain map $\Psi(\mathcal{Z}) \subseteq A$, called **the domain embedding**[4], **is the (linear) representation of a feature field**. Thus, analysis of feature fields within a neural network reduces to analysis of their domain embeddings, as we shall see in subsequent sections. We illustrate the domain embedding of a feature field in Figure 3.

3.2. Linear Field Probes Recover Feature Fields

We introduce *linear field probing* as a method for recovering feature fields from neural activations. Whereas a linear probe identifies the vector that represents a feature variable, a linear field probe identifies the domain embedding that represents a feature field.

Definition 5 *A* ***linear field probe*** *for a feature field $F_{\mathcal{Z}}$ is a family of linear probes $\ell_z : \mathcal{A} \to \mathbb{R}$, with $\ell_z(a) := \langle a, \hat{\Psi}(z)\rangle_{\mathcal{A}}$ for $z \in \mathcal{Z}$, parameterized by a domain map $\hat{\Psi} : \mathcal{Z} \to \mathcal{A}$, and used to approximate the field values via $\ell_z(\Phi(x)) = \langle \Phi(x), \hat{\Psi}(z)\rangle_{\mathcal{A}} \approx f(x, z)$.*

Thus, a feature field is linearly represented in the activations if and only if a linear field probe can recover it from those activations. A field probe is trained by learning a (generally nonlinear[5]) domain map $\hat{\Psi}$. Knowledge of the topology of $\mathcal{Z}$ is not necessarily assumed, so long as a parameterization of $\mathcal{Z}$ is given. For example, a linear field probe may be trained by discretizing $\mathcal{Z}$ and independently training a linear probe ℓ_z for each z.

3.3. Domain Homeomorphism

We have not so far made use of the topology in imposing structure on realizations of the feature field. In the beginning of Section 3, we found that one natural imposition is continuity. For example, our dog tracking bone directions may have similar belief for adjacent directions, so that $f(x, z_{\mathrm{NE}}) \approx f(x, z_{\mathrm{NNE}})$.

Definition 6 *A feature field is called* ***continuous*** *if and only if all realizations $(z \mapsto f(x, z)) \sim F$ are continuous for all $x \sim X$. That is, $f(x, z)$ is continuous in the domain argument z for a fixed data point x.*

We show that, under mild conditions, the domain embedding of continuous feature fields preserves the topology of (i.e. is homeomorphic to, see Appendix A) the domain space itself.

Theorem 7 (Domain Homeomorphism Theorem) *Let $f(x, z) = \langle \Phi(x), \Psi(z)\rangle$ be a linearly represented continuous feature field such that (1) the family $\{f(x, \cdot)\}_{x \in X}$ separates points, (2) $\Phi(X)$ spans the subspace containing $\Psi(Z)$, and (3) Z is compact. Then the domain embedding $\Psi(\mathcal{Z})$ is homeomorphic to the domain space $\mathcal{Z}$.*

These conditions are mild: (1) says distinct domain points are distinguishable (if $z_1 \neq z_2$ then *some* x has $f(x, z_1) \neq f(x, z_2)$), (2) excludes wasted representational capacity, and (3) holds for most natural domains including intervals, circles, and graphs. We defer the proof to Appendix I.

[4] Not to be confused with the activation "manifold" $\Phi(X)$.

[5] As we shall observe in Section 4, linearity in Ψ trivializes the expressivity of the feature field. The linearity of a field probe specifically refers to the linear inner product with the activations $\langle \Phi(x), \hat{\Psi}(z)\rangle_{\mathcal{A}}$.

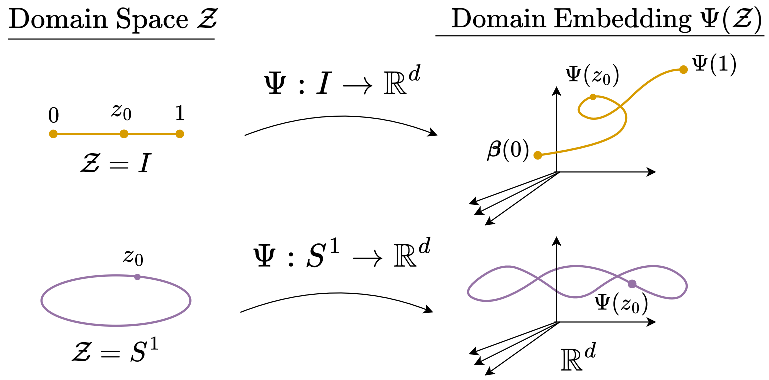
Domain Space $\mathcal{Z}$
Domain Embedding $\Psi(\mathcal{Z})$
$\Psi : I \to \mathbb{R}^d$
0
z_0
1
$\mathcal{Z} = I$
$\Psi(z_0)$
$\Psi(1)$
$\boldsymbol{\beta}(0)$
z_0
$\Psi : S^1 \to \mathbb{R}^d$
$\Psi(z_0)$
$\mathcal{Z} = S^1$
$\mathbb{R}^d$

Figure 3: **Domain Homeomorphism Theorem** (informal): *For continuous feature fields, the domain embedding $\Psi(\mathcal{Z})$ in activation space preserves the topology of the domain space $\mathcal{Z}$, so that $\Psi(\mathcal{Z}) \cong \mathcal{Z}$.*

By the theorem, we have that our dog that tracks directions over the circle $\mathcal{Z} \cong S^1$ represents her beliefs by a circle $\Psi(\mathcal{Z}) \cong S^1$ that resides within her activation space. Thus, she explicitly represents an abstract hidden space that she has never observed directly.

4. The Geometry of Feature Fields

In this section, we show that the geometry of the domain embedding fully determines the space of representable feature fields. The space of square-integrable functions over the domain, $L^2(\mathcal{Z})$, is infinite-dimensional. However, we prove that linearly representable realizations are confined to a finite-dimensional subspace determined entirely by the domain embedding geometry.

Definition 8 *For a feature field $F_{\mathcal{Z}}$ with domain embedding $\Psi(\mathcal{Z})$, the* ***realization space*** *is the space of functions over $\mathcal{Z}$ which are linearly representable realizations of f, given by* $\mathrm{span}(\{f_x(z)\}_{x \in \mathcal{X}})$.

To characterize this space, we define the **feature kernel** $K(z, z') = \langle \Psi(z), \Psi(z') \rangle$, which encodes the geometry of the domain embedding through pairwise inner products. The kernel K defines a function space (called the reproducing kernel Hilbert space, see Appendix A) $\mathcal{H}_K$. We now establish an equivalence between $\mathcal{H}_K$ and the realization space.

Theorem 9 (Field Geometry Equivalence Theorem) *Let $f(x, z) = \langle \Phi(x), \Psi(z) \rangle$ be a linearly represented feature field such that $\Phi(\mathcal{X})$ spans the subspace containing $\Psi(\mathcal{Z})$. Then the realization space is the RKHS associated with the feature kernel, given by*

$$\mathcal{H}_K = \left\{ \sum_{j=1}^{d} \sqrt{\lambda_j} a_j \psi_j(z) : a_j \in \mathbb{R} \right\}. \tag{2}$$

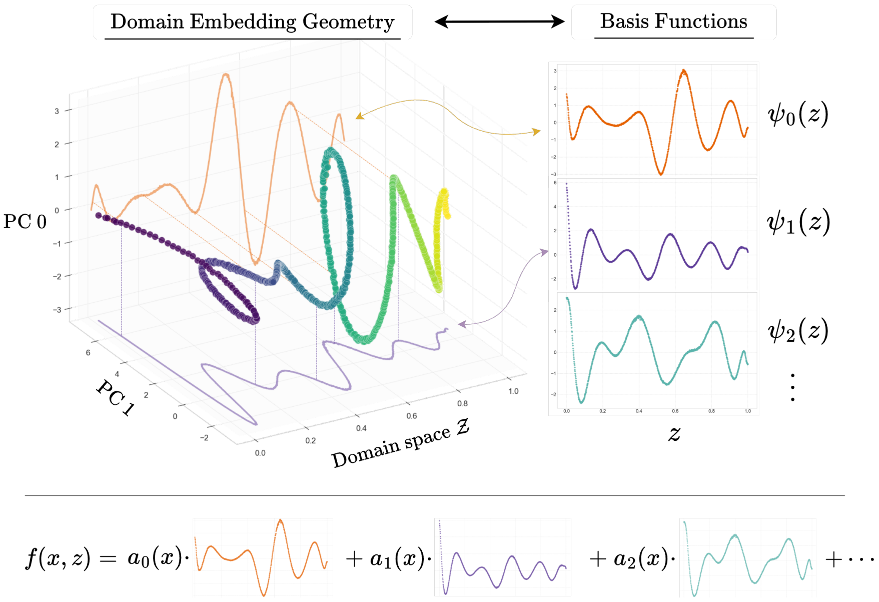

Figure 4: **Field Geometry Equivalence Theorem** (informal): *The geometry of the domain embedding in activation space encodes the basis functions of feature fields.* Shown left is a domain embedding, or *feature manifold*, obtained from a transformer trained on a toy task. Shown right are basis functions which are projections of the embedding geometry.

For any realization $f(x, z)$, there exists a unique spectral representation:

$$f(x, z) = \sum_{j=1}^{d} \sqrt{\lambda_j}\, a_j(x) \psi_j(z) \tag{3}$$

where $a_j(x) = \Phi(x)^T \mathbf{e}_j$ are coefficients determined by projecting the activation onto the principal directions $\{\mathbf{e}_j\}$ of the domain embedding, $\{\psi_j\}$ are the eigenfunctions of the integral operator T_K, and $\{\lambda_j\}$ are the corresponding eigenvalues.

The theorem establishes a direct correspondence between the geometry of the domain embedding and the function space structure of representable fields. The eigenfunctions are given by projecting the domain embedding onto principal directions $\psi_j(z) = \frac{1}{\sqrt{\lambda_j}} \langle \Psi(z), \mathbf{e}_j \rangle$. Since the domain embedding lives in $\mathbb{R}^d$, at most d eigenvalues are nonzero, so $\dim(\mathcal{H}_K) \leq d$. This is a significant constraint: out of the infinite-dimensional space $L^2(\mathcal{Z})$, only a finite-dimensional subspace can be linearly represented, and its basis is encoded by the principal components of the domain embedding.

5. Experiments

In this section, we provide empirical validation for the theoretical framework of feature fields, linear field probing, and domain embedding geometry developed in this paper.

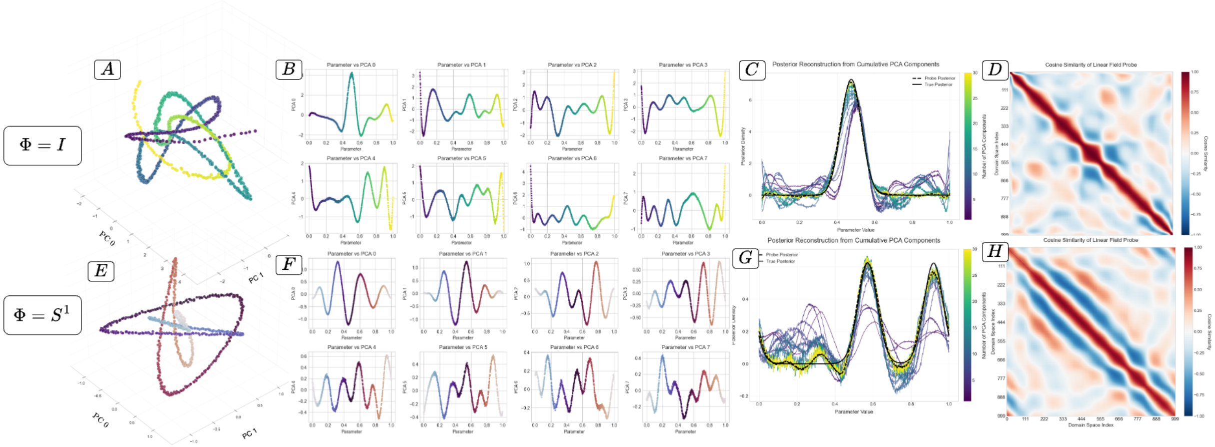

Figure 5: Analysis of feature fields on two topologies. **A,E**: Domain embeddings $\boldsymbol{\mathcal{Z}}$ are homeomorphic to underlying domain spaces. **B,F**: Eigenfunction bases are recovered as neural networks representations of the feature field. **C,G**: Feature field realizations corresponding to data samples, as encoded by basis functions, which become progressively more accurate with more basis functions. **D,H**: Domain embedding kernel, from which basis functions may be obtained.

We experiment with two tasks designed to induce the emergent formation of feature fields of two topologies $\mathcal{Z} = I$ and $\mathcal{Z} = S^1$ inside neural networks. Both tasks are of the following form: *Given parameters of a hidden feature field, output an integral of the feature field.* Mathematically, choose a task $y = m(x)$ where x parametrizes a feature field $f(x, z)$—where for fixed x, $f_x(z)$ is a function over a topology $\mathcal{Z}$—and y is an integral of f, defined as $y = \int_{\mathcal{Z}} f(x, z)g(z)d\mu(z)$. We describe choices of f and g in Appendix C.

We find that feature fields are linearly represented in transformers. We train a linear field probe on transformers trained on the two tasks. The local interpretation allows us to approximate a linear field probe by discretizing $\mathcal{Z}$ and training a linear probe for each point in the discretized domain space. We find that for both tasks, the feature fields are linearly represented in the second layer of a 2-layer transformer but not a 1-layer transformer. Example feature field realizations are displayed in Figure 5 (C and G), Figure 7, and Figure 11.

After training $\Psi(\mathcal{Z})$, the feature kernel was obtained by assembling a Gram matrix $K(z, z') = \langle \Psi(z), \Psi(z') \rangle_A$, displayed in Figure 5 (D and H). Then the eigenfunction basis is obtained by an eigendecomposition of the kernel matrix displayed in Figure 5 (B and F) and Figure 10. Equivalently, the eigenfunction basis is obtained by plotting the projection of $\Psi(z)$ onto its principal components as a function of z. Moreover, for samples of the feature field, we show that the basis function coefficients obtained from the activations in the domain embedding subspace (by projecting the activation onto the principal components of $\Psi(z)$) match those obtained by numerically calculating the integral $\int f(x, z)\psi_i(z)d\mu(z)$ in Figure 8 and Figure 9. These results show that the feature field function basis may be recovered and that the coefficients in neural network activation space match the theoretical coefficients in the function space.

We study training dynamics on the beta distribution over $\mathcal{Z} = I$, and study the development of its geometry over training. In Figure 6, we plot a comparison over the same training run between (1) the effective dimension of the domain embedding in the first and second layers and (2) the linear field probe recovery loss in the first and second layers, along with (3) the transformer task loss. The effective dimension is measured with the participation ratio, defined as $\mathrm{PR} = (\sum_i \lambda_i)^2/(\sum_i \lambda_i^2)$, where $\{\lambda_i\}$ are eigenvalues. We observe a "phase transition" in the task loss as well as in the effective dimension dimension of the domain embedding in the second layer over the same period from roughly 400–600 epochs. We observe the linear field probe loss plateaus near the end of the phase transition period. Additionally, we find little change in the probe loss in the first layer, indicating that the feature field representation emerges in the second layer of the transformer. These findings are consistent with the theoretical findings of Section 4, establishing a correspondence between domain embedding dimension and representational capacity.

6. Related Work

Elhage et al. (2022) considered sparse features with correlated co-occurrence, rather than correlated values; the feature scalars are still independent. Engels et al. (2025) study *multi-dimensional* features, especially features which appear to have circular geometry—like days of the week, or months of the year—and find that projections of the features into their principal components show clean circles. On the other hand, the circular domain embedding in Figure 5 appears highly curved and not so cleanly circular in any two principal components. A clean geometry would result in limited representational capacity. Li et al. (2023) trained nonlinear classifier probes over the 8×8 tiled topology of the Othello board, finding non-trivial geometry in the embedding of the underlying board. This may be considered a non-linear classifier field probe, as the topology of the board was taken into account, although only a preliminary analysis of the geometry was performed. Nanda et al. (2023) performed a follow up analysis, training a linear classifier field probes, although no study of the geometry was performed.

7. Conclusion

We generalize the concept of scalar-valued feature variables to function-valued feature fields. Feature fields are defined over a topological domain space, which encodes the structural relationships of the world. We introduce linear field probes that can extract feature fields from neural networks by discovering an embedding of the topological domain space. We show that, for continuous feature fields, the domain embedding retains the topological structure of the original domain. The geometry of the domain embedding is a kernel that defines the space of representable realizations of the feature field. We validate our claims empirically and find that representations of feature fields naturally emerge in transformer neural networks. This work lays the groundwork for a principled understanding of topological representations, providing a step toward a more complete theory of world models.

References

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

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear, 2025. URL https://arxiv.org/abs/2405.14860.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ICLR*, 2023.

Yuxiao Li, Eric J Michaud, David D Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4):344, 2025.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024a.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024b. URL https://arxiv.org/abs/2311.03658.

Adam Shai, Lucas Teixeira, Alexander Oldenziel, Sarah Marzen, and Paul Riechers. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034, 2024.

Appendix A. Background

Topology A *topological space* is a set T equipped with a collection of open sets that defines a notion of continuity and neighborhood. A map $f : T \to T'$ between topological spaces is said to be *continuous* if the preimage of every open set in T' is open in T. A *homeomorphism* is a bijective, continuous map with a continuous inverse—formally identifying two topological spaces as topologically equivalent. A map $\Psi : T \to \mathbb{R}^d$ is a *topological embedding* if it is continuous, injective, and its image $\Psi(T) \subset \mathbb{R}^d$ inherits the topology of T; that is, Ψ is a homeomorphism onto its image.

Random Fields A *random field* generalizes a scalar-valued random variable to a function-valued random variable defined over a topological space. More formally, given a probability space, a random field F is a collection of random variables $F = \{F_t : t \in T\}$ indexed by points in a topological space T. A sample or *realization* of F is a deterministic function $f : T \to \mathbb{R}$, obtained by evaluating each F_t at the same outcome in the underlying probability space. Continuity of realizations is often assumed to preserve the structure imposed by the topology of T.

Function Spaces A *function space* is a vector space whose elements are functions $f : T \to \mathbb{R}$, defined on a topological domain T. Of particular interest is the space $\mathbb{R}^T$, denoting the set of all real-valued functions on T. Within $\mathbb{R}^T$, a notable subspace is the *Reproducing Kernel Hilbert Space (RKHS)*, a Hilbert space of functions associated with a symmetric, positive-definite kernel $K : T \times T \to \mathbb{R}$. This kernel defines an inner product structure and induces a canonical feature map $t \mapsto K(t, \cdot)$, embedding each point $t \in T$ into the RKHS. The RKHS has the important property that pointwise evaluation is continuous—a consequence of the *reproducing property* $f(t) = \langle f, K(t, \cdot) \rangle$.

Reproducing Kernel Hilbert Spaces (RKHS) Let K be a symmetric and positive definite kernel. According to the Moore-Aronszajn theorem, the kernel K uniquely defines a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$. This kernel defines a linear integral transform $T_K : L^2(\mathcal{Z}) \to L^2(\mathcal{Z})$:

$$[T_K g](z) = \int_{\mathcal{Z}} K(z, z') g(z') d\mu(z') \tag{4}$$

By Mercer's theorem, the kernel K has a spectral decomposition:

$$K(z, z') = \sum_{j=1}^{\infty} \lambda_j \psi_j(z) \psi_j(z') \tag{5}$$

where $\{\psi_j\}$ are the orthonormal eigenfunctions of T_K with corresponding non-negative eigenvalues $\{\lambda_j\}$.

Appendix B. Feature Field Examples

We provide two motivating examples of feature fields below.

1. *Posterior distributions*: Bayesian inference involves updating posterior distributions over entire parameter spaces rather than just tracking the maximum a posterior estimate. A neural network Bayesian receives an observation of the world x and updates a posterior distribution $p(z|x)$ conditioned on the observation over a parameter space topology which encodes the relationship between parameters $\theta \in \Theta$. Thus, we obtain a feature field $f(x, \theta) = p(\theta|x)$ defined over a topology $\mathcal{Z} = \Theta$.

2. *Value functions:* An agent tracks a value function over all possible states and actions, rather than just the highest value state and action. A neural network agent experiences a history x and updates a value function $V_x(s)$ over a state space topology which encodes the relationships between states $s \in S$. Thus, we obtain a feature field $f(x, s) = V_x(s)$ defined over a topology $\mathcal{Z} = S$.

Appendix C. Feature Field Construction

$\mathcal{Z} = I$ We consider a beta distribution $\text{Beta}(a, b)$ feature field defined over the unit interval $I = [0, 1]$. The feature field is defined using the beta distribution probability density function $f = (z; a, b)$,[6] which takes two parameters $a, b > 0$. Our task is for a model to take input $x = (a, b)$ and output a scalar functional $m(x) = \int_0^1 f(x, z)g(z)d\mu(z)$. We choose $g(z; \omega) = \cos(2\pi\omega z)$, with frequency hyperparameter ω.

$\mathcal{Z} = S^1$ We consider a k-mixture of von Mises distributions feature field defined over the circle topology S^1. The feature field is defined with two parameters $x = (\mu_1, \mu_2)$, specifying the means of two von Mises distributions. Then the task is defined as $f(x, z) \propto e^{\kappa \cos(x-\mu_1)} + e^{\kappa \cos(x-\mu_2)}$, $g(z; \omega_1, \omega_2, \omega_3, c_1, c_2, c_3) = c_1 \cos(2\pi\omega_1 z) + c_2 \cos(2\pi\omega_2 z) + c_3 \cos(2\pi\omega_3 z)$.

[6] Notated a, b instead of the typical α, β to not confuse with activations and features.

Appendix D. Notation

Notation	**Meaning / Type**
$f(X)$	Feature (random variable over random input X)
$f(x)$	Feature scalar (value of the feature at a fixed input x)
$f(X, \mathcal{Z})$	Feature field (random field over manifold)
$f(X, z)$	Feature (random variable at a fixed manifold point z)
$f(x, \mathcal{Z})$	Feature function (deterministic function over manifold with fixed input x)
$f(x, z)$	Feature scalar (scalar at fixed input x and fixed manifold point z)
$\Phi(x)$	Activations
A	Activation space
X	Data space
$x \in X$	Data point
$\mathcal{Z}$	Feature space
$z \in \mathcal{Z}$	Domain point
$\Psi(z)$	Feature map
$f(x, z)$	Feature field
$\hat{f}(x, z)$	Density/superposition field
$K(z, z')$	Reproducing kernel
T_K	Kernel transform
$\{\psi_j\}$	Kernel eigenfunctions
$\{\lambda_j\}$	Kernel eigenvalues
$\{a_j\}$	Eigenfunction coefficients, spectral coordinates
$\mathcal{H}_K$	Reproducing kernel Hilbert space

Table 1: Summary of feature notation and their corresponding object types.

Appendix E. Experimental Details

All experiments were conducted on a MacBook Pro with an Apple M1 chip and 16 GB of RAM.

Component	**Setting**
Residual stream dimension	512
Discretization points (linear field probe)	1000
Train/test split (transformer dataset)	60% / 40%
Train/test split (linear field probe dataset)	90% / 10%
Optimizer	Adam (both models)
Training epochs (transformer)	2000
Training epochs (linear field probe)	2000
Attention heads (transformer)	8
Feature Field Functions	
Beta distribution:	$g(z) = \cos(3.5 \cdot 2\pi z)$
Von Mises mixture:	$g(z) = 100\cos(2\pi z) + 50\cos(4\pi z - 0.71)$ $+25\cos(6\pi z - 0.123)$
Von Mises concentration (κ)	10

Table 2: Summary of experimental hyperparameters and functional forms used in field probing.

Appendix F. Equations

$$\begin{aligned}
\Phi(x) &= \int_{\mathcal{Z}} \hat{f}(x,z)\Psi(z)d\mu(z) = T_{\Psi}[\hat{f}_x] = \sum_{j=1}^{d} a_j \boldsymbol{e}_j \\
\Psi(z) &= \sum_{j=1}^{d} \sqrt{\lambda_j}\psi_j(z)\boldsymbol{e}_j = \mathcal{Z}(K^{1/2}) \\
K(z,z') &= \sum_{j=1}^{d} \lambda_j \psi_j(z)\psi_j(z') = \langle \Psi(z), \Psi(z') \rangle_A \\
K^{1/2}(z,z') &= \sum_{j=1}^{d} \sqrt{\lambda_j}\psi_j(z)\psi_j(z') \\
[T_K g](z) &= \int_{\mathcal{Z}} K(z,z')g(z')d\mu(z') = \langle g, K_z \rangle_{L^2} \\
T_{K^{1/2}} &= \mathcal{Z} \circ T_{\Psi} \\
[T_{\Psi} g] &= \int_{\mathcal{Z}} g\Psi(z)d\mu(z) \\
f(x,z) &= \sum_{j=1}^{d} \sqrt{\lambda_j} a_j(x)\psi_j(z) = \langle \boldsymbol{\Phi}(x), \Psi(z) \rangle_A = \langle f_x, K_z \rangle_{\mathcal{H}_K} \\
\hat{f}(x,z) &= T_K^{-1}[f] = \sum_{j=1}^{d} \frac{1}{\sqrt{\lambda_j}} a_j(x)\psi_j(z)
\end{aligned}$$

Appendix G. Training Effective Dimension

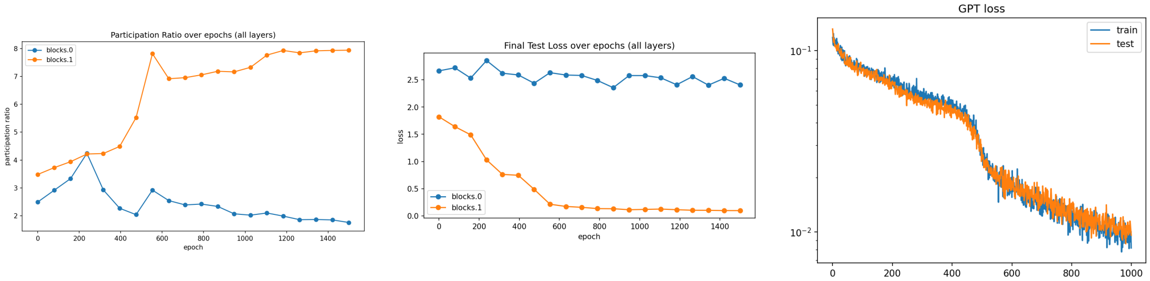

Figure 6: We plot the effective dimension of the domain embedding, approximated with the participation ratio, across layers and training. We observe a phase transition in the second layer embedding (left) which corresponds to the phase transition in the model loss curve (right), both between 400 and 600 epochs. After this period, both the domain embedding effective dimension and the linear field probe loss plateaus. Meanwhile, the first layer linear field probe shows no evidence for the development of linear representation of the feature field.

Appendix H. Linear Field Probe Results

In the following figures, we display feature field realizations, an eigenfunction basis, and coefficients, which were extracted through the use of linear field probes.

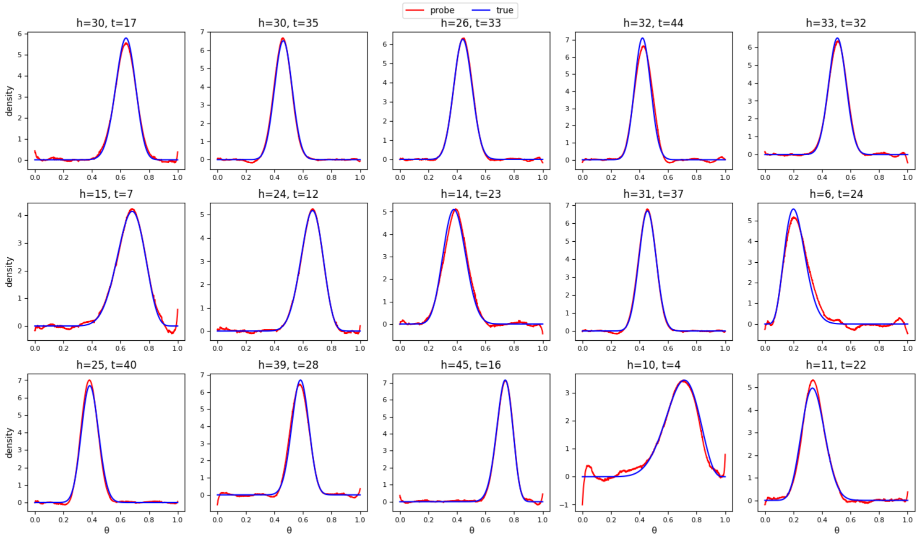

Figure 7: 9 realizations of the beta distribution obtained from transformer activations using the linear field probe.

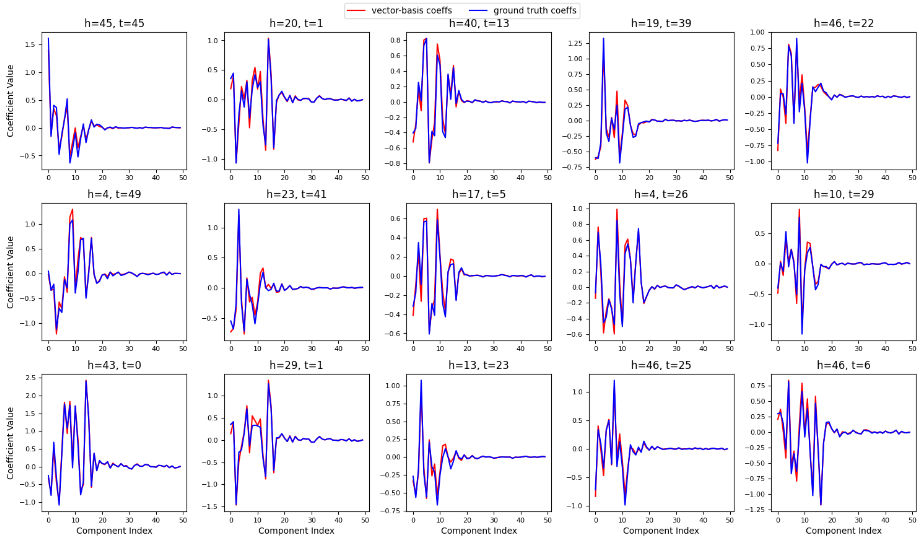

Figure 8: For the beta distribution feature field, eigenbasis coefficients match between eigenfunctions and closed form integral.

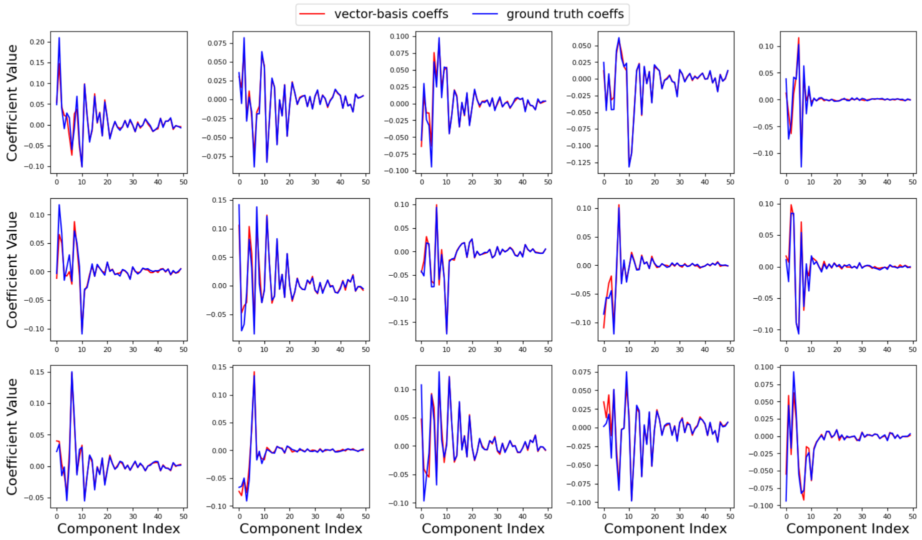

Figure 9: For the von Mises mixture feature field, eigenbasis coefficients match between eigenfunctions and closed form integral.

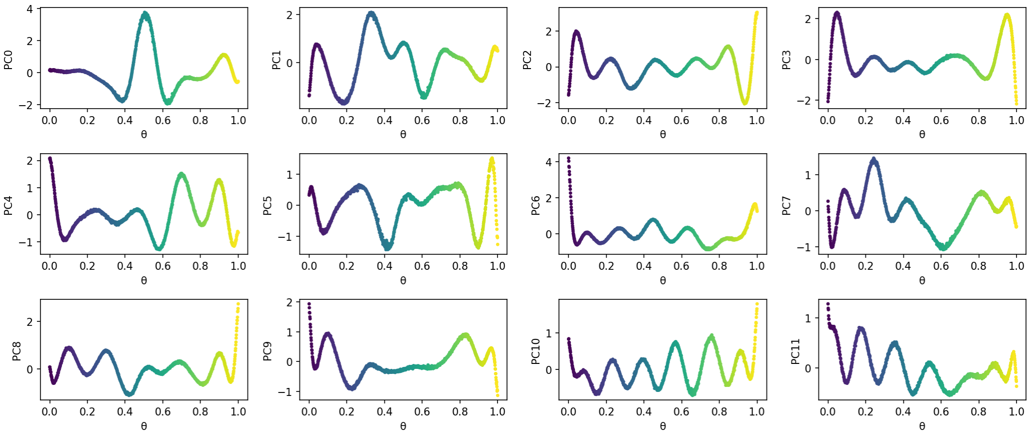

Figure 10: First 12 eigenfunction basis for beta distribution feature field.

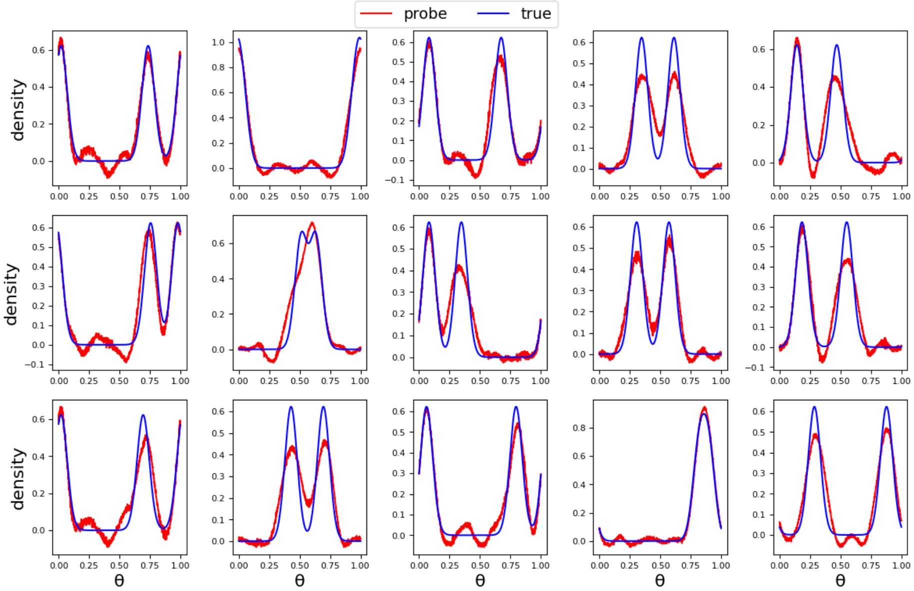

Figure 11: Fifteen realizations of the mixture von Mises distribution obtained from transformer activations using the linear field probe.

Appendix I. Proofs

We provide proofs for the two theorems of the paper.

I.1. Theorem 1

Theorem 10 (Domain Homeomorphism Theorem) *Let $f(x,z) = \langle \Phi(x), \Psi(z) \rangle$ be a linearly represented continuous feature field satisfying:*

1. *For each $x \in X$, the map $z \mapsto f(x,z)$ is continuous.*
2. *The family $\{f(x,\cdot)\}_{x\in X}$ separates points: if $z_1 \neq z_2$, then $f(x,z_1) \neq f(x,z_2)$ for some $x \in X$.*
3. *The set $\Phi(X)$ spans the subspace containing $\Psi(Z)$.*
4. *Z is compact.*

Then $\Psi(\mathcal{Z})$ is homeomorphic to $\mathcal{Z}$.

Proof We show that Ψ is a continuous bijection onto its image. Since Z is compact and $\mathbb{R}^d$ is Hausdorff, this implies Ψ is a topological embedding.

Step 1: Ψ is injective.
Suppose $\Psi(z_1) = \Psi(z_2)$. Then for all $x \in X$,

$$f(x,z_1) = \langle \Phi(x), \Psi(z_1) \rangle = \langle \Phi(x), \Psi(z_2) \rangle = f(x,z_2).$$

Since the family $\{f(x,\cdot)\}_{x\in X}$ separates points, we conclude $z_1 = z_2$.

Step 2: Ψ is continuous.
Let $V = \operatorname{span}\{\Psi(z) : z \in Z\}$. By assumption, $\Phi(X)$ spans V, so we can choose $x_1, \ldots, x_k \in X$ such that $\{\Phi(x_1), \ldots, \Phi(x_k)\}$ is a basis for V.

Define the map

$$g : Z \to \mathbb{R}^k, \quad g(z) = \big(f(x_1,z), \ldots, f(x_k,z)\big)^\top.$$

Since each $f(x_i,\cdot)$ is continuous, g is continuous.

Now, write $\Psi(z) = \sum_{j=1}^k c_j(z)\, \Phi(x_j)$ for some coefficients $c_j(z)$. Then

$$g(z)_i = f(x_i,z) = \langle \Phi(x_i), \Psi(z) \rangle = \sum_{j=1}^k c_j(z)\, \langle \Phi(x_i), \Phi(x_j) \rangle = [G\, c(z)]_i,$$

where G is the Gram matrix with entries $G_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$. Since $\{\Phi(x_i)\}$ is linearly independent, G is invertible, so $c(z) = G^{-1} g(z)$ and

$$\Psi(z) = \sum_{i=1}^k [G^{-1} g(z)]_i\, \Phi(x_i).$$

This expresses Ψ as a composition of continuous maps, so Ψ is continuous.

Step 3: Conclusion.
We have shown that $\Psi : Z \to \Psi(Z)$ is a continuous bijection. Since Z is compact and $\Psi(Z) \subseteq \mathbb{R}^d$ is Hausdorff, Ψ is a topological embedding. So $\Psi(\mathcal{Z})$ is homeomorphic to $\mathcal{Z}$. ■

I.2. Theorem 9 (Field Geometry Equivalence Theorem)

Theorem 11 (Field Geometry Equivalence Theorem) *Let $f(x, z) = \langle \Phi(x), \Psi(z) \rangle$ be a linearly represented feature field such that $\Phi(\mathcal{X})$ spans the subspace $V = \operatorname{span}\{\Psi(z) : z \in \mathcal{Z}\}$. Then the realization space equals the RKHS $\mathcal{H}_K$ associated with the feature kernel $K(z, z') = \langle \Psi(z), \Psi(z') \rangle$.*

Proof We prove the theorem in three steps: (1) every realization f_x lies in $\mathcal{H}_K$, (2) the realization space equals $\mathcal{H}_K$, and (3) the spectral representation holds.

Step 1: Every realization lies in $\mathcal{H}_K$.

Let $V = \operatorname{span}\{\Psi(z) : z \in \mathcal{Z}\}$. By assumption, $\Phi(\mathcal{X})$ spans V, so we may write any $\Phi(x)$ as:

$$\Phi(x) = \sum_{i=1}^{n} c_i \Psi(z_i) + \Phi(x)^\perp$$

where $\Phi(x)^\perp \perp V$. Since $\Psi(z) \in V$ for all z, the orthogonal component does not contribute:

$$f_x(z) = \langle \Phi(x), \Psi(z) \rangle = \sum_{i=1}^{n} c_i \langle \Psi(z_i), \Psi(z) \rangle = \sum_{i=1}^{n} c_i K(z_i, z)$$

Thus f_x is a finite linear combination of kernel sections $K_{z_i}(\cdot) = K(z_i, \cdot)$, which lies in $\mathcal{H}_K$ by definition of the RKHS.

Step 2: The realization space equals $\mathcal{H}_K$.

From Step 1, every $f_x \in \mathcal{H}_K$, so $\operatorname{span}\{f_x : x \in \mathcal{X}\} \subseteq \mathcal{H}_K$.

For the reverse inclusion, we show every kernel section K_{z_0} lies in the realization space. Fix $z_0 \in \mathcal{Z}$. Since $\Psi(z_0) \in V = \operatorname{span}\{\Phi(x) : x \in \mathcal{X}\}$, there exist $x_1, \ldots, x_n \in \mathcal{X}$ and coefficients $c_1, \ldots, c_n$ such that:

$$\Psi(z_0) = \sum_{i=1}^{n} c_i \Phi(x_i)$$

Then for any $z \in \mathcal{Z}$:

$$K_{z_0}(z) = \langle \Psi(z_0), \Psi(z) \rangle = \sum_{i=1}^{n} c_i \langle \Phi(x_i), \Psi(z) \rangle = \sum_{i=1}^{n} c_i f_{x_i}(z)$$

Hence $K_{z_0} \in \operatorname{span}\{f_x\}$. Since kernel sections span $\mathcal{H}_K$, we have $\mathcal{H}_K \subseteq \operatorname{span}\{f_x : x \in \mathcal{X}\}$.

Step 3: The spectral representation.

Let $C = \int_{\mathcal{Z}} \Psi(z)\Psi(z)^T d\mu(z)$ be the covariance matrix of the domain embedding, with eigendecomposition $C = \sum_j \lambda_j \mathbf{e}_j \mathbf{e}_j^T$ where $\{\mathbf{e}_j\}$ are orthonormal eigenvectors and $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$.

Define $\psi_j(z) = \langle \Psi(z), \mathbf{e}_j \rangle / \sqrt{\lambda_j}$ for each j with $\lambda_j > 0$.

Claim: $\{\psi_j\}$ *are orthonormal in* $L^2(\mathcal{Z}, \mu)$.

$$\begin{aligned}
\langle \psi_j, \psi_k \rangle_{L^2} &= \frac{1}{\sqrt{\lambda_j \lambda_k}} \int_{\mathcal{Z}} \langle \Psi(z), \mathbf{e}_j \rangle \langle \Psi(z), \mathbf{e}_k \rangle \, d\mu(z) \\
&= \frac{1}{\sqrt{\lambda_j \lambda_k}} \mathbf{e}_j^T \left(\int_{\mathcal{Z}} \Psi(z) \Psi(z)^T d\mu(z) \right) \mathbf{e}_k \\
&= \frac{1}{\sqrt{\lambda_j \lambda_k}} \mathbf{e}_j^T C \, \mathbf{e}_k = \frac{\lambda_k}{\sqrt{\lambda_j \lambda_k}} \delta_{jk} = \delta_{jk}
\end{aligned}$$

Claim: $\{\psi_j\}$ *are eigenfunctions of* T_K *with eigenvalues* $\{\lambda_j\}$.

The domain embedding can be written as $\Psi(z) = \sum_j \sqrt{\lambda_j} \psi_j(z) \mathbf{e}_j$, which gives the Mercer decomposition:

$$K(z, z') = \langle \Psi(z), \Psi(z') \rangle = \sum_j \lambda_j \psi_j(z) \psi_j(z')$$

Applying T_K to ψ_k:

$$[T_K \psi_k](z) = \int_{\mathcal{Z}} K(z, z') \psi_k(z') d\mu(z') = \sum_j \lambda_j \psi_j(z) \langle \psi_j, \psi_k \rangle_{L^2} = \lambda_k \psi_k(z)$$

Claim: The spectral representation holds.

$$f(x, z) = \langle \Phi(x), \Psi(z) \rangle = \left\langle \Phi(x), \sum_j \sqrt{\lambda_j} \psi_j(z) \mathbf{e}_j \right\rangle = \sum_j \sqrt{\lambda_j} \, \underbrace{\langle \Phi(x), \mathbf{e}_j \rangle}_{= a_j(x)} \, \psi_j(z)$$

This completes the proof. ■