Proceedings Track

Neural Manifold Geometry Encodes Feature Fields

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 representation, Interpretability, Linear probe, World models

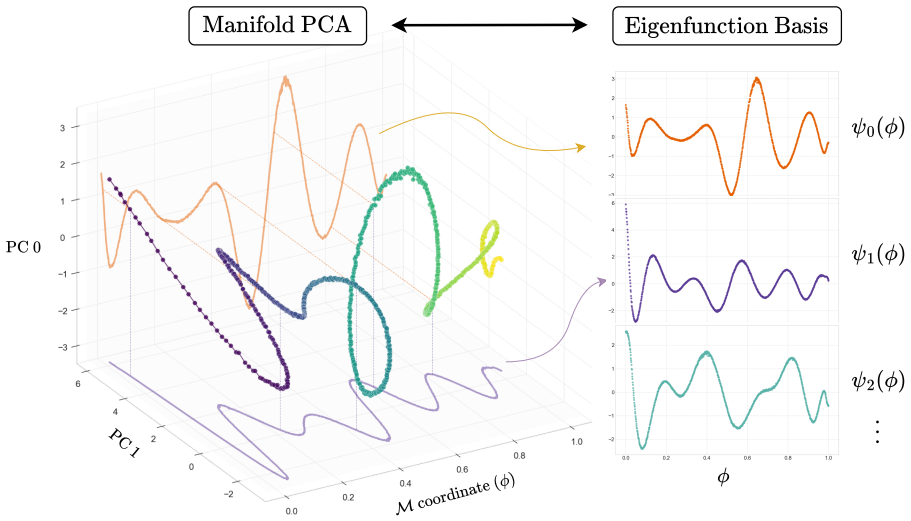

Figure 1: **Field Geometry Equivalence Theorem** (informal): *The geometry of the domain embedding in activation space encodes the basis functions of feature fields.* Shown left is a domain embedding, or *feature manifold*, obtained from a transformer trained on a toy task. Shown right are basis functions which are projections of the embedding geometry.

1. Introduction

The rapid advancement of artificial neural networks has continued alongside substantial institutional investment, whilst a principled understanding of their internal workings remains elusive. 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.

In particular, much effort has been made in uncovering the hypothesized atoms, or *features* of neural representation, typically assuming these features to be independent or *separable* Elhage et al. (2022); Park et al. (2024b). When these features have been associated to points in activation space (see Section 2), researchers have revealed the geometry appears to be highly structured Engels et al. (2025); Li et al. (2025); Park et al. (2024a); Shai et al. (2024); Li et al. (2023). Motivated by the relational structure such geometry may encode, we introduce *feature fields*, a representation whose atoms are intrinsically related by their correspondence to an underlying topological space, as illustrated in Figure 2.

We find that feature fields are represented in neural networks as an embedding of their topology into activation space, as illustrated in Figure 4. In the special case where the topology is a manifold, we believe that these *domain embeddings* may be related to the "feature manifolds" speculated about in the literature Olah (2024). Furthermore, the geometry of this embedding fully defines the space of linearly representable functions for a given feature field, as shown in Figure 1. Feature fields are a generalization of the atomic picture of features: by considering a single domain point, the atomic features are recovered; however the structure of the topology is forgotten.

In this paper, we make the following contributions: In Section 2, we define previous atomic conceptions of features as *feature variables*. In Section 3, we formalize *feature fields*, generalizing feature variables, which encode structural dependencies by the topology of the domain space over which they are defined. In Section 4, we introduce linear field probing as a method for recovering feature fields by discovering a domain embedding. In Section 5, we prove that continuous linear fields are represented with a topological embedding of their domain space in activation dual space. In Section 6, we prove that feature fields live in a finite-dimensional Hilbert space, whose basis functions are encoded by the domain embedding geometry. Finally, in Section 7, we trace the emergence and geometric evolution of domain embeddings on several topologies across layers and training in transformers.

2. Feature Variables

A straightforward way to determine whether a neural network represents a concept is by examining scalar "features." Given an input space (e.g. an image) and a concept (e.g. "food"), a feature variable is a function that maps each input to a scalar value corresponding to a specific concept. As an example, consider a hypothetical dog neural network. The input distribution contains the set of images that the dog sees, and the potential features include "how bone-like" or "which compass direction ϕ".

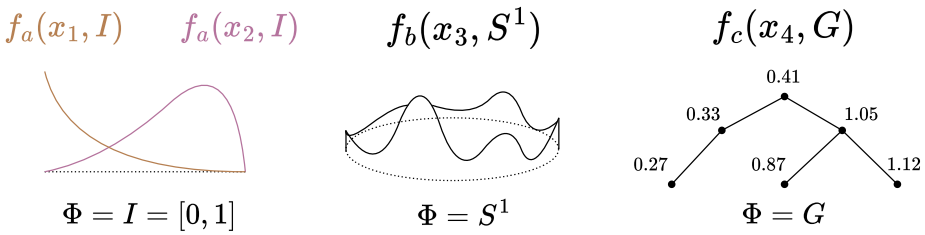

Figure 2: An illustration of three feature fields on distinct topologies: the interval I, the circle S^1, and the graph G. The first feature field shows two different realizations of the same feature field. For every data point $x \sim X$, field maps a data point to a function over the domain space: $\phi \mapsto f(x, \phi) \in \mathbb{R}^\Phi$. For example, the above feature fields may represent 1) posterior distributions over parameter spaces; 2) value function over a state space.

Definition 1 *A **feature variable** F_i associated with a concept i is a scalar-valued random variable defined by applying a deterministic, scalar function f_i to the input random variable X, i.e. $f_i : X \to \mathbb{R}$, with $x \mapsto f_i(x)$. For each input $x \in X$, the scalar value $f_i(x)$ is called the **feature scalar**.*

Note that this definition does not reference the neural network internals. Quantifying how much the network represents a given feature typically involves training a probe model, such as the linear probe we discuss below. In Section 4, we generalize this approach to linear field probes to detect structure beyond just scalars.

Linear Probing We say a neural network *represents* a feature variable X if the network's activations can be used to recover the corresponding feature scalar $f_i(x)$ for a given input $x \in X$. These activations reside in a d-dimensional *activation space* $A \subset \mathbb{R}^d$, where the network maps each input $x \in X$ to an activation vector $\boldsymbol{\alpha}(x) \in A$.

Linear probing is a popular method to recover feature variables from network activations Alain and Bengio (2016); Belinkov (2022). A linear probe for a feature variable F_i is a linear map $P_i : A \to \mathbb{R}$ approximating the feature scalar as $f_i(x) \approx P_i(\boldsymbol{\alpha}(x)) := \langle \boldsymbol{\alpha}(x), \boldsymbol{\beta}_i \rangle_A$, where $\boldsymbol{\beta}_i \in \mathbb{R}^d$ is a trainable weight vector.[1] The inner product on activation space A is the standard Euclidean dot product augmented with a constant offset $\langle \boldsymbol{v}, \boldsymbol{w} \rangle_A = \boldsymbol{v} \cdot \boldsymbol{w} + w_d$.[2] Whether a feature variable is represented by a neural network is naturally defined by whether a linear probe can reconstruct it from activations:

Definition 2 *A feature variable F_i is **linearly represented** if and only if there exists a weight vector $\boldsymbol{\beta}_i \in \mathbb{R}^d$ of F_i, such that $f_i(x) = \langle \boldsymbol{\alpha}(x), \boldsymbol{\beta}_i \rangle_A$ for all $x \in X$.*

In Figure 3, we illustrate linear probes within activation space and contrast them with *linear field probes*, introduced later in Section 4.

1. Technically, $\boldsymbol{\beta}_i \in A^* \subset \mathbb{R}^d$ is a vector from the dual activation space A^*.
2. Equivalently, we can augment the activation vector with a constant-valued dimension and incorporate the offset term into the weight vector.

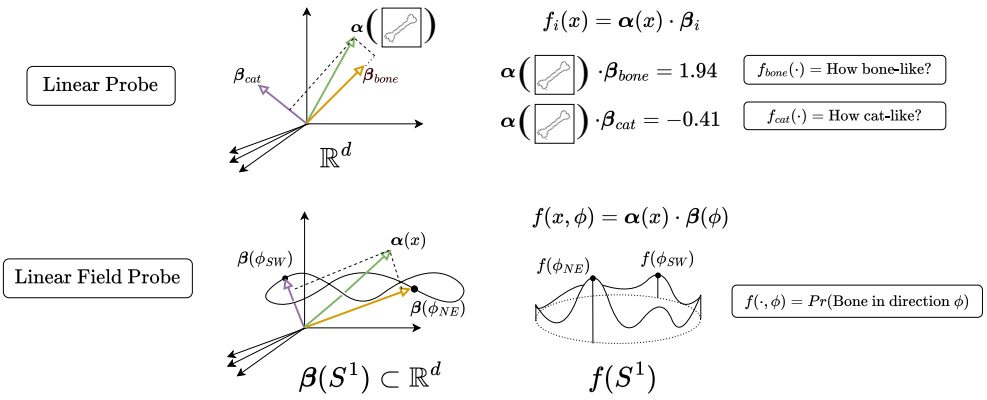

Figure 3: An illustration of feature variables and feature field for a hypothetical dog neural network. A linear probe associates a feature scalar $f_i(x)$ with a vector $\boldsymbol{\beta}_i$ with $f_i(x) \approx \boldsymbol{\alpha}(x) \cdot \boldsymbol{\beta}_i$ for activation $\boldsymbol{\alpha}(x)$. We introduce the *linear field probe* which associates a *feature field* $f(x, \phi)$ with a vector field $\boldsymbol{\beta}(\phi)$with $f(x, \phi) \approx \boldsymbol{\alpha}(x) \cdot \boldsymbol{\beta}(\phi)$. We illustrate an example of a linear field probe of a feature field over S^1, where a neural network represents the probability of facing toward any orientation ϕ including northeast ϕ_{NE} and southwest ϕ_{SW}.

3. Feature Fields Are Topological Representations

Practical neural networks rarely represent just one scalar feature variable. As a result, prior work often considers collections of distinct feature variables. For example, one can define a collection of feature variables corresponding to the probability of various events (e.g. "finding a bone", "finding a ball", "finding nothing") in some event space ("outcomes of digging a hole"). This approach is practical as long as there are few enough events (i.e. feature variables).

However, in real-world settings, treating feature variables as distinct ignores important structural dependencies between them. Suppose we consider a different distribution where events and their corresponding probabilities are related (e.g. "in which direction did I bury the bone?"). Neighboring features variables (directions) have correlated probabilities values, but the collection of distinct features treats them as independent. We wish to know not only whether neural networks represent the individual features, but also whether they encode the structural relationships between them.

We therefore introduce a class of topologically structured representations that naturally capture such interdependencies, which we call *feature fields*.

Definition 3 *A **feature field** F over a topological space Φ (the **domain space**) is a random field, induced by a deterministic function*

$$f : X \times \Phi \to \mathbb{R}, \quad (x, \phi) \mapsto f(x, \phi),$$

such that, to every data point $x \sim X$, the feature field assigns the scalar function

$$F(x) := (\phi \mapsto f(x, \phi)) \in \mathbb{R}^{\Phi}.$$

called the ***realization*** *of the feature field at x.*

Feature fields have a *global* and *local* interpretation which we describe in Appendix B alongside motivating examples. Feature fields are intrinsically topological representations explicitly connected to the structured domain space on which they are defined. This domain space is *latent* or *hidden* in the sense of not being directly observable by the neural network through the data X: the domain space exists *out there in the world*.

4. Linear Field Probes Extract Feature Fields

We introduce *linear field probing* as a method for extracting feature fields from neural activations.

Definition 4 *A* ***linear field probe*** *for a feature field f is a function $P : A \times \Phi \to \mathbb{R}$ approximating f, defined by*

$$f(x, \phi) \approx P(\boldsymbol{\alpha}(x), \phi) = \langle \boldsymbol{\alpha}(x), \boldsymbol{\beta}(\phi) \rangle_A, \tag{1}$$

where P is parameterized by a ***domain map*** *$\boldsymbol{\beta} : \Phi \to A^*$.*

A linear field probe inherits global and local interpretations from the feature field it approximates. For a fixed data point $x \sim X$, it provides a global view as a function over the domain space, $(\phi \mapsto P(\boldsymbol{\alpha}(x), \phi)) \in \mathbb{R}^{\Phi}$. Conversely, fixing a domain parameter ϕ yields a local linear probe P_ϕ corresponding to the feature variable F_ϕ. Thus, a linear field probe may be understood as a parameterized family of linear probes. Figure 3 illustrates this relationship.

It is important to note that the domain map $\boldsymbol{\beta}$ need not be linear—indeed, linearity would trivialize the geometry, thus trivializing the representational capacity, as we will see in Section 6. The linearity of a field probe specifically refers to the linear inner product used to extract the field from activations.

5. The Domain Embedding

We now turn to the central object of study for the rest of the paper: the *domain embedding*. We show that this object serves as the natural representation of feature fields. Furthermore, for continuous feature fields, the domain embedding in activation space is topologically isomorphic (homeomorphic) to the original domain space.

Whether a feature field is represented by a model is naturally defined by whether it is extractable from the activations by a linear field probe:

Definition 5 *A feature field is* ***linearly represented*** *if and only if there exists a domain map $\boldsymbol{\beta}$ such that*

$$f(x, \phi) = \langle \boldsymbol{\alpha}(x), \boldsymbol{\beta}(\phi) \rangle_A \quad \text{for all } x \in X, \phi \in \Phi. \tag{2}$$

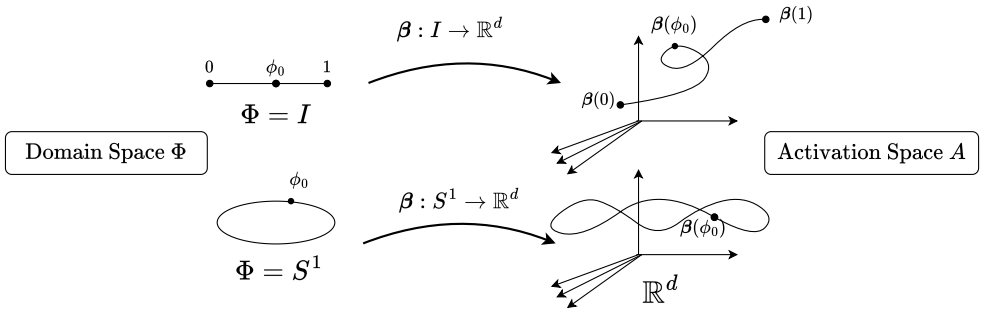

Figure 4: **Domain Homeomorphism Theorem**: *For continuous feature fields, the domain map to its image in activation space $\boldsymbol{\beta} : \Phi \to \boldsymbol{\beta}(\Phi) \subset \mathbb{R}^d$ is a homeomorphism. Equivalently, the domain embedding $\boldsymbol{\beta}(\Phi)$ is a topological embedding.*

Under this definition, the image $\boldsymbol{\beta}(\Phi) \subset A$, called the *domain embedding*, is a representation of the feature field. In contrast, a feature variable is represented by a single point in activation space. Note the difference between domain embeddings $\boldsymbol{\beta}(\Phi)$ and the activation "manifold" $\boldsymbol{\alpha}(X)$: Although both live in activation space[3], the activation manifold is an embedding the data space X, whereas domain embedding is an embedding of the domain space Φ. We illustrate the domain embedding of a feature field in Figure 4.

For the rest of the paper, we turn our attention to continuous feature fields:

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

We now show that the feature fields representation, the domain embedding $\boldsymbol{\beta}(\Phi)$ in activation space, is homeomorphic to the domain space (we defer all proofs to the appendix):

Theorem 7 (Domain Homeomorphism Theorem) *Consider a linearly represented feature field $f(x, \phi) = \langle \boldsymbol{\alpha}(x), \boldsymbol{\beta}(\phi) \rangle$. Suppose that for each fixed x, the map $\phi \mapsto f(x, \phi)$ is continuous and injective, and that the set $\boldsymbol{\alpha}(X)$ spans the space containing $\boldsymbol{\beta}(\Phi)$. Then, the embedding $\boldsymbol{\beta} : \Phi \to \mathbb{R}^d$ is a homeomorphism to its image.*

Thus, the domain embedding $\boldsymbol{\beta}(\Phi)$ is a topological embedding, i.e., it retains the topology of the original domain space Φ. The remarkable implication is: **A neural network which represents a feature field holds a topologically intact image of the "latent" domain Φ which it has never "seen" directly**. This domain Φ is latent in the sense of *hidden* to the neural network, which has only ever directly observed X.

6. The Geometry of Feature Fields

In this section, we show feature fields are confined to live in a particular reproducing kernel Hilbert space (RKHS) which is determined by the geometry of the feature manifold em-

3. Technically, $\boldsymbol{\beta}(\Phi)$ resides in the dual space A^*

bedding $\boldsymbol{\beta}(\mathcal{M})$. The space of square-integrable fields over the manifold is given by $L^2(\Phi)$, forming an infinite dimensional function space. However, we find that representable feature fields live in a much smaller, finite dimensional RKHS.

Definition 8 *For a feature field $f(x, \phi)$, the **feature field space** is a space of deterministic functions over Φ which are linearly representable realizations of f given by $\{f_x(\phi) \mid x \in X\}$*

Define the *feature kernel* $K(\phi, \phi') = \langle \boldsymbol{\beta}(\phi), \boldsymbol{\beta}(\phi') \rangle$ encoding the geometry of the feature manifold embedding. K is continuous, symmetric, and positive definite by construction, and therefore uniquely defines a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$. We now show an equivalence between the feature field space and $\mathcal{H}_K$:

Theorem 9 (Field Geometry Equivalence Theorem) *Feature field space is the RKHS associated with the feature kernel given by*

$$\mathcal{H}_K = \left\{ \sum_{j=1}^{d} \sqrt{\lambda_j} a_j \psi_j(\phi) : f \in L^2(\mathcal{M}) \right\}. \tag{3}$$

For any linear feature field $f(x, \phi)$, there exists a unique representation:

$$f(x, \phi) = \sum_{j=1}^{d} \sqrt{\lambda_j} a_j(x) \psi_j(\phi) \tag{4}$$

where $a_j(x) = \boldsymbol{\alpha}(x)^T \boldsymbol{e}_j$ are coefficients determined by projecting the activation, $\{\psi_j\}$ are the eigenfunctions of T_K, and $\{\lambda_j\}$ are the corresponding eigenvalues.

This establishes that the geometry of the feature embedding encodes the representation of the feature space. Equivalently, the principal projections of the feature embedding are identical to the eigenfunction basis of the feature field.

We visualize Theorem 2 in Figure 1. The feature field space is a strict subset of the space of all square-integrable functions on the feature manifold $\mathcal{H}_K \subset L^2(\Phi)$. $L^2(\Phi)$ is an infinite dimensional function space. Since our feature map is finite-dimensional (living in $\mathbb{R}^d$), only the first d eigenvalues are non-zero, and so $\mathcal{H}_K$ may be at most d dimensional. Thus, we establish that the domain embedding directly defines the feature field space.

7. 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.

7.1. Feature Field Tasks

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

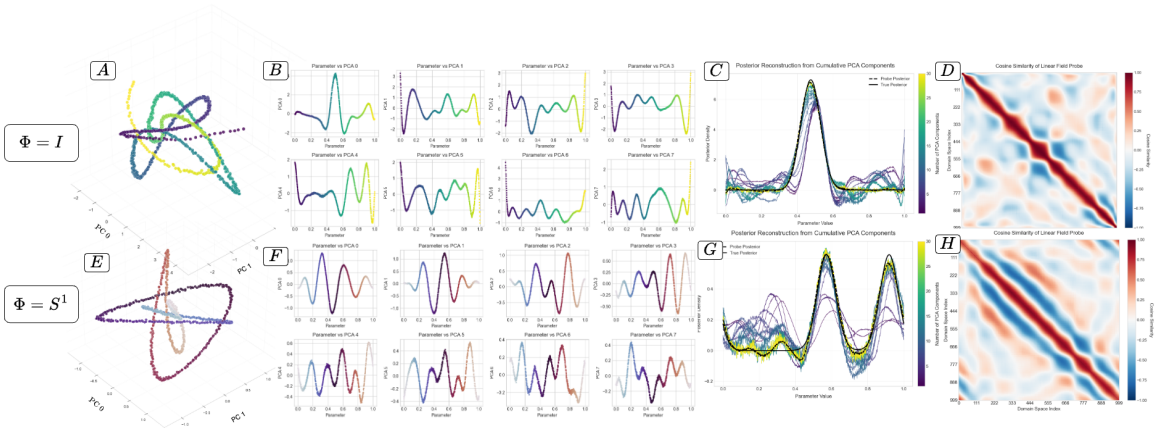

Figure 5: Analysis of feature fields on two topologies. **A,E**: Domain embeddings $\mathbf{\Phi}$ are homeomorphic to underlying domain spaces. **B,F**: Eigenfunction bases are recovered as neural networks representations of the feature field. **C,G**: Feature field realizations corresponding to data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

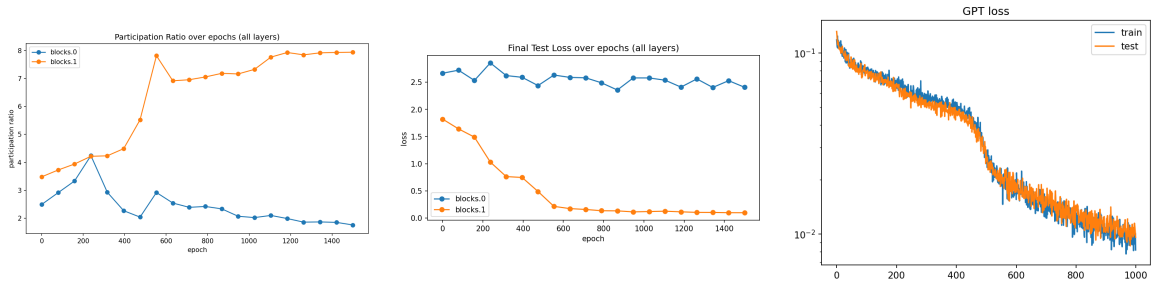

Figure 6: We plot the effective dimension of the domain embedding, approximated with the participation ratio, across layers and training. We observe a phase transition in the second layer embedding (left) which corresponds to the phase transition in the model loss curve (right), both between 400 and 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.

7.2. Extracting Feature Fields with Linear Field Probes

We find that feature fields are linearly represented in transformers. We train a linear field probe on transformers trained on the two tasks defined in Section 7.1. The local interpretation allows us to approximate a linear field probe by discretizing Φ 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.

7.3. Extracting the RKHS with the Domain Embedding

The discretized linear field probes trained in Section 7.2 are parameterized with a discretized matrix encoding $\boldsymbol{\beta}(\Phi)$. The kernel is obtained assembling a Gram matrix $K(\phi, \phi') = \langle \boldsymbol{\beta}(\phi), \boldsymbol{\beta}(\phi') \rangle_A$, 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 $\boldsymbol{\beta}(\phi)$ onto its principal components as a function of ϕ. 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 $\boldsymbol{\beta}(\phi)$) match those obtained by numerically calculating the integral $\int f(x, \phi)\psi_i(\phi)d\mu(\phi)$ 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.

7.4. Training Dynamics

We study training dynamics on the beta distribution over $\Phi = I$, 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, 2) the linear field probe recovery loss in the first and second layers, and 3) the transformer task loss. The effective dimension is measured with the participation ratio, defined as $PR = (\sum_i \lambda_i^2)/(\sum_i \lambda_i^2)$, 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 6, establishing a correspondence between domain embedding dimension and representational capacity.

8. 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, which 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.

Chris Olah. What is a linear representation? what is a multidimensional feature? https://transformer-circuits.pub/2024/july-update/index.html#linear-representations, July 2024. Transformer Circuits Thread. Accessed 29 April 2025.

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 $\beta : T \to \mathbb{R}^d$ is a *topological embedding* if it is continuous, injective, and its image $\beta(T) \subset \mathbb{R}^d$ inherits the topology of T; that is, β 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* $f(t) = \langle f, K(t, \cdot)\rangle$. We provide a detailed discussion of RKHS theory in Appendix J.

Appendix B. Feature Field Interpretations and Examples

Interpretations We can better understand a feature field by viewing it from two perspectives, each offering different insights into its structure and behavior.

- **Global Interpretation:** Fixing a data point $x \in X$ gives a *global* view of the feature field. In this view, the feature field maps a data point to a function over the domain space: $\phi \mapsto f(x, \phi) \in \mathbb{R}^\Phi$. In contrast, a feature variable maps a data point to a scalar value.
- **Local Interpretation:** Fixing a domain point $\phi \in \Phi$ gives a *local* view of the feature field. In this view, each domain point defines a distinct feature variable ($x \mapsto f(x, \phi)$) that assigns a scalar value to each data point. Here, the feature field is naturally seen as a collection of interrelated feature variables $F_\phi : \phi \in \Phi$, indexed by the topology. An equivalent definition consistent with this idea is provided in Appendix I.

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 $p(\phi|x)$ 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 $\Phi = \Theta$.

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 $\Phi = S$.

Appendix C. Feature Field Construction

$\Phi = I$ We consider a beta distribution 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 $f = (\phi; a, b)$[4], which takes two parameters $a, b > 0$. Our task is for a model to take input $x = (a, b)$ and output a scalar functional $m(x) = \int_0^1 f(x, \phi)g(\phi)d\mu(\phi)$. We choose $g(\phi; \omega) = \cos(2\pi\omega\phi)$, with frequency hyperparameter ω.

$\Phi = S^1$ 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 distribution. Then the task is defined as $f(x, \phi) \propto e^{\kappa \cos(x-\mu_1)} + e^{\kappa \cos(x-\mu_2)}$, $g(\phi; \omega_1, \omega_2, \omega_3, c_1, c_2, c_3) = c_1 \cos(2\pi\omega_1\phi) + c_2 \cos(2\pi\omega_2\phi) + c_3 \cos(2\pi\omega_3\phi)$.

Appendix D. Related Work on Features

Correlated features Elhage et al. (2022) considered sparse features with correlated co-occurrence, rather than correlated values; the feature scalars are still independent.

Multidimensional features Engels et al. (2025) study *irreducible* features, especially features which appear to have circular geometry, like days of the week, or months of the year. Projections of the features into the 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. Future work may investigate the relationship between feature fields and multidimensional features.

Existing examples of feature fields Li et al. (2023) trained a nonlinear classifier probes over the eight-by-eight 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.

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

Appendix E. 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)
$f(X, \mathcal{M})$	Feature field (random field over manifold)
$f(X, \phi)$	Feature (random variable at a fixed manifold point ϕ)
$f(x, \mathcal{M})$	Feature function (deterministic function over manifold with fixed input x)
$f(x, \phi)$	Feature scalar (scalar at fixed input x and fixed manifold point ϕ)
$\alpha(x)$	Activations
A	Activation space
X	Data space
$x \in X$	Data point
$\mathcal{M}$	Feature space
$\theta \in \mathcal{M}$	Domain point
$\beta(\phi)$	Feature map
$f(x, \phi)$	Feature field
$\hat{f}(x, \phi)$	Density/superposition field
$K(\phi, \phi')$	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 F. 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:	$g(\phi) = \cos(3.5 \cdot 2\pi\phi)$
Von Mises mixture:	$g(\phi) = 100\cos(2\pi\phi) + 50\cos(4\pi\phi - 0.71) + 25\cos(6\pi\phi - 0.123)$
Von Mises concentration (κ)	10

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

Appendix G. Equations

$$\alpha(x) = \int_{\mathcal{M}} \hat{f}(x,\phi)\beta(\phi)d\mu(\phi) = T_\beta[\hat{f}_x] = \sum_{j=1}^{d} a_j e_j$$

$$\beta(\phi) = \sum_{j=1}^{d} \sqrt{\lambda_j}\psi_j(\phi)e_j = \Phi(K^{1/2})$$

$$K(\phi,\phi') = \sum_{j=1}^{d} \lambda_j \psi_j(\phi)\psi_j(\phi') = \langle \beta(\phi), \beta(\phi')\rangle_A$$

$$K^{1/2}(\phi,\phi') = \sum_{j=1}^{d} \sqrt{\lambda_j}\psi_j(\phi)\psi_j(\phi')$$

$$[T_K g](\phi) = \int_{\mathcal{M}} K(\phi,\phi')g(\phi')d\mu(\phi') = \langle g, K_\phi\rangle_{L^2}$$

$$T_{K^{1/2}} = \Phi \circ T_\beta$$

$$[T_\beta g] = \int_{\mathcal{M}} g\beta(\phi)d\mu(\phi)$$

$$f(x,\phi) = \sum_{j=1}^{d} \sqrt{\lambda_j} a_j(x)\psi_j(\phi) = \langle \alpha(x), \beta(\phi)\rangle_A = \langle f_x, K_\phi\rangle_{\mathcal{H}_K}$$

$$\hat{f}(x,\phi) = T_K^{-1}[f] = \sum_{j=1}^{d} \frac{1}{\sqrt{\lambda_j}} a_j(x)\psi_j(\phi)$$

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.

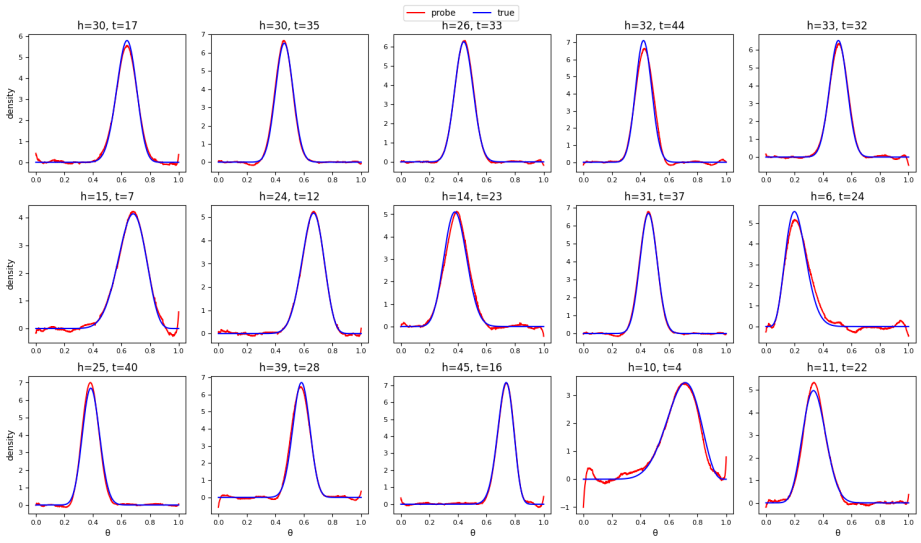

Figure 7: 9 realizations of the beta distribution obtained from transformer activations using the linear field probe.

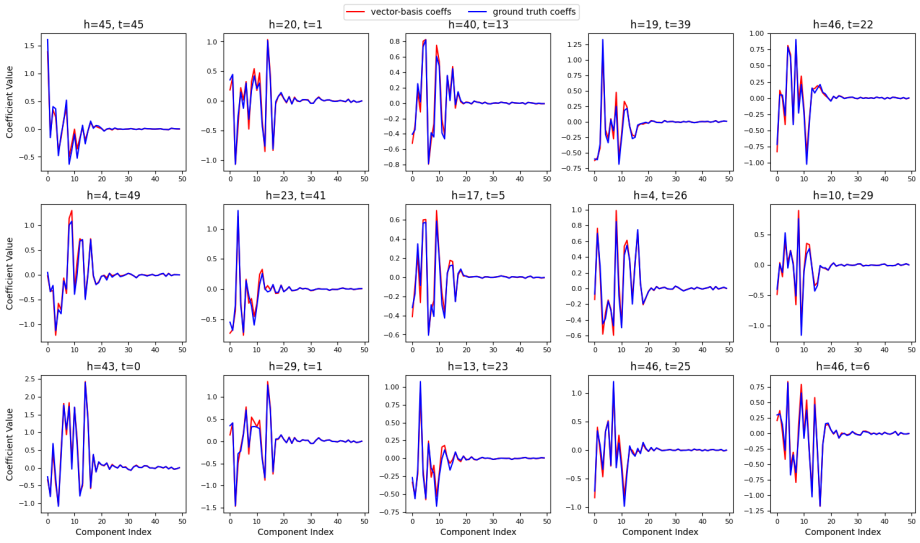

Figure 8: For the beta distribution feature field, eigenbasis coefficients match between eigenfunctions and closed form integral.

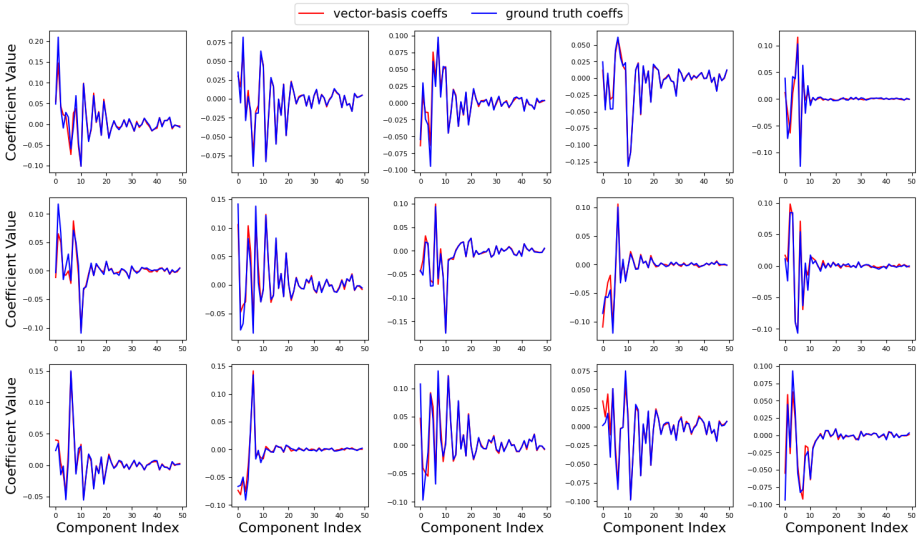

Figure 9: For the von mises mixture feature field, eigenbasis coefficients match between eigenfunctions and closed form integral.

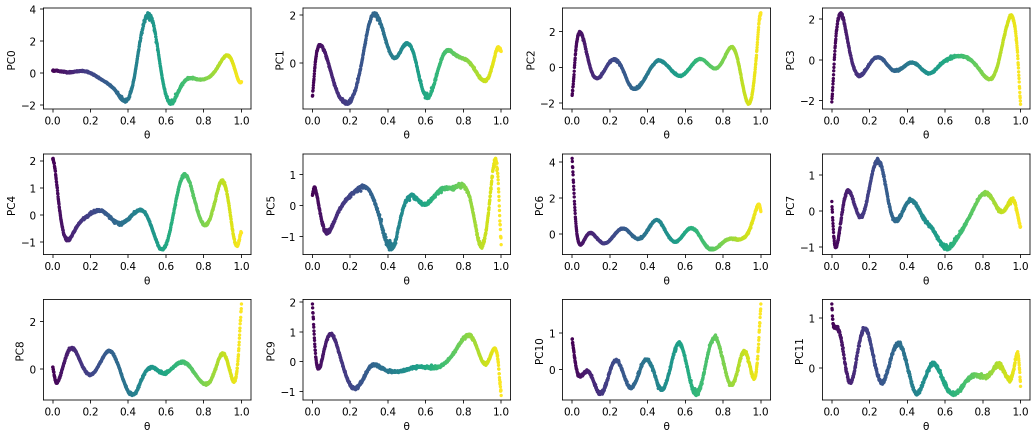

Figure 10: First 12 eigenfunction basis for beta distribution feature field.

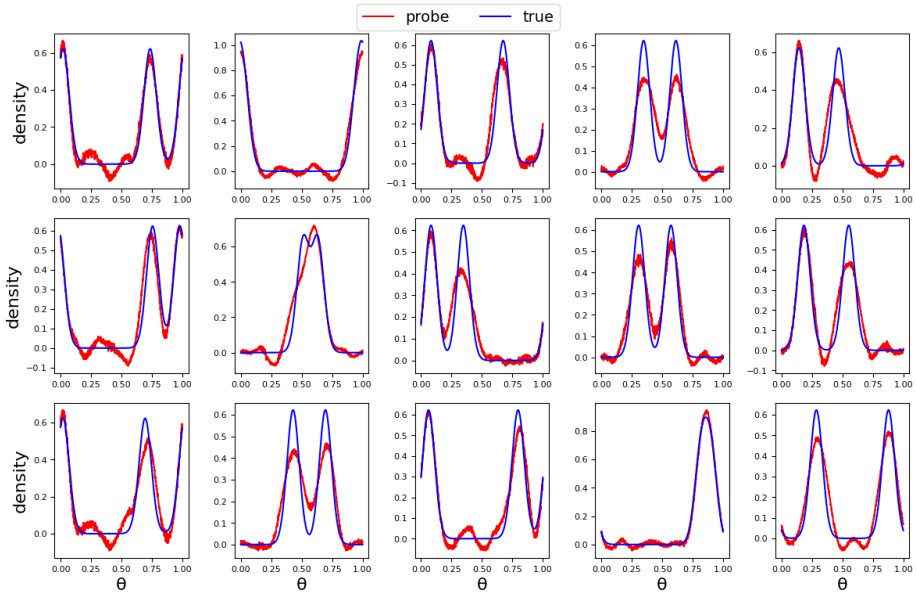

Figure 11: Fifteen realizations of the mixture von Mises distribution obtained from transformer activations using the linear field probe.

Appendix I. Feature Field Alternative Definition

Definition 10 (Feature field) *A* feature field F_T *a collection of feature variables* $\{F_t : t \in T\}$*, where every point in the topology* t *is associated with a feature variable* F_t*.*

Appendix J. Reproducing Kernel Hilbert Spaces

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 $T_K : L^2(\mathcal{M}) \to L^2(\mathcal{M})$:

$$[T_K g](\phi) = \int_{\mathcal{M}} K(\phi, \phi') g(\phi') d\mu(\phi') \tag{5}$$

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

$$K(\phi, \phi') = \sum_{j=1}^{\infty} \lambda_j \psi_j(\phi) \psi_j(\phi') \tag{6}$$

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

Appendix K. Non-orthogonal features have correlated measurements

Consider two features scalars f_a, f_b by with associated unit vectors, $v_a, v_b, |v_i| = 1$. For simplicity let $f_a, f_b \sim N(0, 1)$. Assume that $p(f_a, f_b) = p(f_a)p(f_b)$ features are uncorrelated, and $v_a \cdot v_b = c > 0$ feature vectors are not orthogonal.

Let the activation be a superposition of these two vectors $x = f_a v_a + f_b v_b$

Then the feature measurements are $\langle f_a \rangle = v_a \cdot x = f_a + f_b(v_a \cdot v_b)$

The measurement correlation of the features $\rho(\langle f_a \rangle, \langle f_b \rangle)$ is then

$$\begin{aligned}
Cov(\langle f_a \rangle, \langle f_b \rangle) &= \mathbb{E}[\langle f_a \rangle \langle f_a \rangle] - \mathbb{E}[\langle f_a \rangle]\mathbb{E}[\langle f_b \rangle] \\
&= \mathbb{E}[c f_a^2 + c f_b^2 + (1 + c^2) f_a f_b] = 2c^2 \\
Var(\langle f_a \rangle) &= \mathbb{E}[f_a + c f_b] = 1 + c^2 \\
Var(\langle f_b \rangle) &= \mathbb{E}[f_b + c f_c] = 1 + c^2 \\
\rho(\langle f_a \rangle, \langle f_b \rangle) &= Cov(\langle f_a \rangle, \langle f_b \rangle) / \sqrt{Var(\langle f_a \rangle) Var(\langle f_b \rangle)} \\
&= \frac{2c}{1 + c^2}
\end{aligned}$$

where we see now that the measurements are correlated for $c > 0$.

Appendix L. Discrete features as special case

For discrete non-orthogonal features, the machinery developed in this paper still applies, accounting for the geometry in a way that handles correlations rather than trying to force low coherence.

Appendix M. Proofs

We provide proofs for the two theorems of the paper.

M.1. Theorem 1

Proof Let A be a finite-dimensional real inner–product space and write

$$f(x,\phi) \;=\; \langle \alpha(x), \beta(\phi) \rangle_A, \qquad x \in X,\ \phi \in \Phi.$$

Step 0. The relevant subspace. Set

$$V \;=\; \operatorname{span}\big\{\beta(\phi) \mid \phi \in \Phi\big\} \subseteq A.$$

By hypothesis the family $\{\alpha(x)\}_{x\in X}$ also spans V. Restrict the inner product of A to V; henceforth every vector lives in V. Because V is finite-dimensional, choose points $x_1, \ldots, x_k \in X$ such that

$$B \;=\; \big\{\alpha(x_1), \ldots, \alpha(x_k)\big\}$$

is a basis of V.

1. Injectivity of β. Suppose $\beta(\phi_1) = \beta(\phi_2)$. Then $f(x,\phi_1) = f(x,\phi_2)$ for every $x \in X$, contradicting the assumed injectivity of $f(x,\cdot)$. Thus β is injective.

2. Continuity of β. Form the *Gram matrix* $G \in \mathbb{R}^{k\times k}$ with entries $G_{ij} = \langle \alpha(x_i), \alpha(x_j) \rangle$. Because B is linearly independent, G is symmetric positive-definite and hence invertible.

Define the map

$$g : \Phi \longrightarrow \mathbb{R}^k, \qquad g(\phi) = \big(f(x_1,\phi), \ldots, f(x_k,\phi)\big)^\top.$$

Each component $f(x_i,\cdot)$ is continuous, so g is continuous. Write any vector $v \in V$ in the basis B as $v = \sum_{i=1}^k c_i(v)\, \alpha(x_i)$ and note $g(\phi) = G\, c\big(\beta(\phi)\big)$. Hence

$$c\big(\beta(\phi)\big) = G^{-1} g(\phi), \qquad \beta(\phi) = \sum_{i=1}^k \big[G^{-1} g(\phi)\big]_i\, \alpha(x_i).$$

Because g and matrix multiplication by G^{-1} are continuous, β is continuous.

3. Continuity of the inverse. Give $\beta(\Phi) \subseteq V$ the subspace topology. Let $(y_\lambda) \subseteq \beta(\Phi)$ converge to $y \in \beta(\Phi)$ and set $\phi_\lambda = \beta^{-1}(y_\lambda)$, $\phi = \beta^{-1}(y)$. For any $x \in X$ the inner product is continuous, so

$$f(x,\phi_\lambda) = \langle \alpha(x), y_\lambda \rangle \;\longrightarrow\; \langle \alpha(x), y \rangle = f(x,\phi).$$

Thus $f(x,\phi_\lambda) \to f(x,\phi)$ for *every* $x \in X$. Because each map $f(x,\cdot)$ is injective, the family $\{f(x,\cdot)\}_{x\in X}$ separates points of Φ; therefore the only possible limit for the net (ϕ_λ) is ϕ, i.e. $\phi_\lambda \to \phi$. Hence β^{-1} is continuous.

4. Conclusion. $\beta\colon \Phi \to \beta(\Phi)$ is bijective, continuous, and has a continuous inverse, so it is a *homeomorphism onto its image*. Consequently $\beta(\Phi)$ is a topological embedding of the domain space Φ. ∎

M.2. Theorem 2

Let Φ be a topological space and V a (real) Hilbert space with inner product $\langle \cdot, \cdot \rangle_V$. Suppose we are given two maps

$$\alpha : X \longrightarrow V, \qquad \beta : \Phi \longrightarrow V,$$

and define the *feature field*

$$f : X \times \Phi \;\longrightarrow\; \mathbb{R}, \qquad f(x, \phi) \;=\; \big\langle \alpha(x), \beta(\phi) \big\rangle_V.$$

Then, for each fixed $x \in X$, the section $f_x : \Phi \to \mathbb{R}$ given by $f_x(\phi) = f(x, \phi)$ belongs to a reproducing-kernel Hilbert space on Φ.

Proof 1. Build the kernel from the embedding. Define

$$K(\phi, \psi) \;=\; \big\langle \beta(\phi), \beta(\psi) \big\rangle_V, \qquad \phi, \psi \in \Phi.$$

For any finite collection $\{\phi_1, \ldots, \phi_m\} \subset \Phi$ and scalars $c_1, \ldots, c_m \in \mathbb{R}$,

$$\sum_{i,j=1}^{m} c_i c_j \, K(\phi_i, \phi_j) = \Big\| \sum_{i=1}^{m} c_i \, \beta(\phi_i) \Big\|_V^2 \;\geq 0,$$

so K is symmetric and positive-definite. By the Moore–Aronszajn theorem there exists a unique RKHS $\mathcal{H}_K \subset \mathbb{R}^\Phi$ whose reproducing kernel is K.

2. Express f_x as a kernel combination. Fix $x \in X$. Because V is a Hilbert space, the vector $\alpha(x)$ can be approximated in norm by finite linear combinations of $\beta(\phi)$:

$$\alpha(x) \;=\; \lim_{m \to \infty} \sum_{j=1}^{m} a_j^{(m)} \, \beta(\phi_j^{(m)}), \qquad a_j^{(m)} \in \mathbb{R}, \; \phi_j^{(m)} \in \Phi.$$

For any finite expansion we obtain

$$\big\langle \alpha(x), \beta(\phi) \big\rangle_V = \sum_{j=1}^{m} a_j^{(m)} \, \big\langle \beta(\phi_j^{(m)}), \beta(\phi) \big\rangle_V = \sum_{j=1}^{m} a_j^{(m)} \, K\Big(\phi_j^{(m)}, \phi \Big) \, .$$

Hence each truncated version of f_x lies in $\mathcal{H}_K$ (finite kernel sums are by definition elements of the RKHS).

3. Take the limit in the RKHS. The space $\mathcal{H}_K$ is complete, and the above finite sums converge in the $\mathcal{H}_K$ norm because

$$\Big\| \sum_{j=1}^{m} a_j^{(m)} K(\phi_j^{(m)}, \cdot) \Big\|_{\mathcal{H}_K} = \Big\| \sum_{j=1}^{m} a_j^{(m)} \, \beta(\phi_j^{(m)}) \Big\|_V \longrightarrow \|\alpha(x)\|_V \;<\; \infty.$$

Thus the exact section f_x is the norm limit of elements already in $\mathcal{H}_K$, so $f_x \in \mathcal{H}_K$. Moreover

$$\|f_x\|_{\mathcal{H}_K} \;=\; \|\alpha(x)\|_V.$$

4. Conclusion. For every $x \in X$ the slice $\phi \mapsto f(x, \phi)$ resides in the RKHS $\mathcal{H}_K$ built from the kernel $K(\phi, \psi) = \langle \beta(\phi), \beta(\psi) \rangle_V$. Hence the given inner-product form guarantees that the feature field lives in a reproducing-kernel Hilbert space. ■