Learning Theory of Transformers: Local-to-Global Approximation via Softmax Partition of Unity

Zhongjie Shi
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, United States
zshi332@gatech.edu

Wenjing Liao
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, United States
wliao60@gatech.edu

Abstract

This paper investigates the learning theory of Transformer networks for regression tasks on the compact Euclidean domain $[0,1]^d$ and d-dimensional compact Riemannian manifolds. We propose a novel constructive approximation framework for Transformers that builds local approximations of the target function and aggregates them into a global approximation via softmax partition of unity. This approach leverages the attention mechanism to achieve spatial localization through affine transformations of the input. The softmax activation plays a crucial role in aggregating local approximations to a global output. From an approximation perspective, we prove that a dense Transformer equipped with only two encoder blocks and standard single-hidden-layer point-wise feed-forward networks can achieve a uniform ε-approximation error for α-Hölder continuous functions with $\alpha \in (0,1]$ using $\mathcal{O}(\varepsilon^{-d/\alpha})$ total parameters. Building upon this approximation guarantee, we establish a near minimax-optimal generalization error bound of order $\mathcal{O}\big(n^{-\frac{2\alpha}{2\alpha+d}}\log n\big)$ for the empirical risk minimizer, where n is the training data size. The Transformer architecture studied in this paper uses native softmax attention and sinusoidal positional encodings, with a fixed depth of two encoder blocks. Finally, numerical experiments reveal spatially localized, Gaussian-like attention weights that emerge during training, resembling our Softmax POU weights.

1 Introduction

During the past decade, deep learning has achieved remarkable success in various domains, including natural language processing [16, 44, 26], computer vision [22, 15, 41], and complex games [40]. As a leading architecture, the Transformer [44] has become central to these advancements: Vision Transformers (ViTs) have surpassed classical convolutional networks in image recognition [10], while large language models (LLMs) demonstrate exceptional generative and reasoning abilities [2, 32]. Despite this widespread empirical success, the theoretical understanding of the expressivity and generalization capability of the Transformers remains limited.

To establish a learning theory of Transformers, extensive research has investigated their expressivity and generalization. Initial studies focused on universal approximation properties, demonstrating that various Transformer models can universally approximate continuous (permutation equivariant) sequence-to-sequence functions [48, 49, 50, 21]. Beyond universality, inspired by the constructive techniques developed for classical deep ReLU networks [46, 33, 38], recent studies have shown that Transformers can achieve satisfying approximation bounds and has the potential to overcome the curse of dimensionality under certain structural assumptions [17, 42, 19, 18, 37, 20]. Based on these approximation results, further studies have provided generalization analysis and statistical rates. These works evaluate sample complexity and estimation errors in different theoretical settings

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

[51, 11, 17, 45, 42, 18, 39, 36, 37]. However, despite these theoretical advances, a clear understanding of the specific role the softmax attention mechanism plays in the expressivity of transformer networks remains elusive, uncovering how this attention mechanism facilitates efficient feature representation learning is still an important problem.

A standard local-to-global approach to deep-ReLU-neural-network approximation on the unit cube $[0,1]^d$ covers the domain with small, geometrically simple patches, typically derived from a Cartesian grid or a triangulation, and combines local approximants through a partition of unity (POU) [46, 47, 33]. Depending on the target's smoothness, the local approximants are constants or Taylor polynomials; representative POU constructions use piecewise-linear spikes on triangulations [47] or tensor products of one-dimensional trapezoidal functions [46, 33]. However, localization in these constructions is tied to prescribed Cartesian grids or simplicial triangulations, whose adaptation to complex geometries such as compact manifolds requires additional geometric constructions. Chen et al. [3, 4] establish manifold approximation results for deep ReLU networks using local coordinate charts and smooth POUs. Havrilla and Liao [18] extend this approach to Transformers with ReLU attention. However, it remains underexplored whether softmax attention can directly construct local-to-global approximations that adapt to complex domain geometries without explicit meshes or local coordinate charts.

We develop a mesh-free local-to-global constructive approximation framework based on Softmax POU, which provides a unified local-to-global approximation framework for Euclidean domains and manifolds, with potential extensions to more general geometries. Our method uses smooth, normalized weights to aggregate local approximations through native softmax attention. Specifically, to approximate a target function $g : \mathcal{X} \to \mathbb{R}$, we cover the input domain $\mathcal{X}$ with ℓ_2 balls of radius r_g centered at a discrete set of points $\{\boldsymbol{c}_i\}_{i=1}^{C_g}$. Given a scaling parameter $M_g > 0$, we define the spatially localized weights as

$$\beta_i(\boldsymbol{x}) = \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_k\|_2^2)\right)} = \frac{\exp\left(2M_g\langle \boldsymbol{x},\boldsymbol{c}_i\rangle - M_g\|\boldsymbol{c}_i\|_2^2\right)}{\sum_{k=1}^{C_g}\exp\left(2M_g\langle \boldsymbol{x},\boldsymbol{c}_k\rangle - M_g\|\boldsymbol{c}_k\|_2^2\right)}. \quad (1)$$

The target function g is then approximated by the convex combination of its values at these discrete centers: $\widehat{g}(\boldsymbol{x}) = \sum_{i=1}^{C_g}\beta_i(\boldsymbol{x})g(\boldsymbol{c}_i)$. This Softmax POU has three properties relevant to the Transformer construction. Geometrically, its weights use Euclidean distances to covering centers without requiring an explicit Cartesian grid, allowing the construction to be applied on manifolds. The normalized weight at each center depends on the distances to all centers, coupling their contributions through a shared denominator. Algebraically, softmax enforces the exact POU condition $\sum_{i=1}^{C_g}\beta_i(\boldsymbol{x}) = 1$, while the common input-dependent quadratic term $\|\boldsymbol{x}\|_2^2$ cancels in the normalization. The resulting scores are affine in $\boldsymbol{x}$ and can be realized through native dot-product attention. These properties provide a direct correspondence between the local-to-global approximation scheme and the Transformer construction.

Building upon these insights, this paper provides a comprehensive theoretical analysis of the expressive power and generalization capabilities of Transformers. Our main contributions are summarized as follows

- **Softmax POU Approximation Framework.** We propose a novel local-to-global constructive approximation framework based on a globally supported Softmax POU. By algebraically canceling input-dependent quadratic terms, this mesh-free POU mechanism translates relative Euclidean distances into purely affine transformations, inherently establishing a dynamic global attention competition to achieve spatial localization. This exact structural correspondence empowers shallow Transformers to seamlessly fuse localized approximations into an aggregated global approximation via native softmax dot-product attention mechanism.
- **Learning on Euclidean Domains via Softmax POU.** Applying this framework to compact Euclidean spaces $[0,1]^d$, we constructively prove that a shallow (only two encoder blocks with standard single-hidden-layer FFNs), wide, and dense Transformer with softmax activation and sinusoidal positional encodings achieves a uniform ε-approximation error for α-Hölder continuous functions using $\mathcal{O}(\varepsilon^{-d/\alpha})$ parameters (see Theorem 1). Building upon this, we establish a near minimax-optimal convergence rate of $\mathcal{O}\big(n^{-\frac{2\alpha}{2\alpha+d}}\log n\big)$ (see Theorem 2), formally validating the inherent statistical efficiency of the native Transformer architecture without relying on artificial sparsity or structural depth constraints.

- **Adaptivity to Riemannian Manifolds.** We extend our theoretical guarantees to target functions defined on d-dimensional compact Riemannian manifolds embedded in $\mathbb{R}^{\bar{d}}$. We formally prove that by operating directly on the ambient Euclidean metric, the Softmax POU framework natively adapts to such complex geometries. Consequently, shallow Transformers can achieve a uniform ε-approximation error using $\mathcal{O}(\varepsilon^{-d/\alpha})$ parameters (see Theorem 3) and a near minimax-optimal convergence rate of $\mathcal{O}\big(n^{-\frac{2\alpha}{2\alpha+d}}\log n\big)$ (see Theorem 4). Because these bounds scale exclusively with the intrinsic dimension d rather than the ambient dimension $\bar{d}$, our analysis establishes the native Transformer architecture's inherent capacity to adapt to complex geometries and overcome the curse of dimensionality.

The remainder of this paper is organized as follows: Section 2 defines the Transformer architecture and its hypothesis space. Section 3 establishes the theoretical guarantees on compact Euclidean domains: Subsection 3.1 introduces the Softmax POU framework to prove uniform approximation bounds, and Subsection 3.2 leverages these results to derive generalization rates. Section 4 extends this framework to compact Riemannian manifolds, demonstrating the architecture's adaptivity to complex geometries and its ability to avoid the curse of dimensionality. Section 5 presents numerical experiments providing empirical support for the spatial localization mechanism underlying our construction. Section 6 discusses related literature, and Section 7 concludes the paper. Appendices A–C provide detailed proofs, and Appendix D provides experimental details and additional results.

Notation. We use lower case bold letters for vectors ($\boldsymbol{x}, \boldsymbol{y}$), upper case bold for matrices ($\boldsymbol{X}, \boldsymbol{A}$), and calligraphic letters for sets and spaces ($\mathcal{M}, \mathcal{H}$). For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_p$ denotes its standard ℓ_p norm, and $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ denotes the Euclidean inner product. For a matrix $\boldsymbol{X}$, $\mathrm{vec}(\boldsymbol{X})$ denotes its column-wise vectorization, $\|\boldsymbol{X}\|_{\max} = \max_{i,j}|\boldsymbol{X}_{i,j}|$ denotes the maximum absolute value among its elements. For a function $f : \mathcal{X} \to \mathbb{R}$, $\|f\|_\infty := \sup_{\boldsymbol{x}\in\mathcal{X}}|f(\boldsymbol{x})|$ denotes its supremum norm. Let $\mathcal{B}(\boldsymbol{c}, r)$ and $\bar{\mathcal{B}}(\boldsymbol{c}, r)$ denote the open and closed Euclidean balls centered at $\boldsymbol{c}$ with radius r, respectively. We define $[n] := \{1, \ldots, n\}$. The Kronecker delta $\delta_{i,j}$ evaluates to 1 if $i = j$ and 0 otherwise.

2 Transformer architecture

In this section, we formally introduce the structure of the Transformer model. The architecture consists of a pre-processing stage followed by L blocks of Transformer encoders. Each encoder block comprises a multi-head attention (MHA) layer, followed by a point-wise FFN layer.

For an input vector $\boldsymbol{x} \in \mathbb{R}^d$, the pre-processing step $\mathcal{P} : \mathbb{R}^d \to \mathbb{R}^{D\times P}$ converts the input to an initial hidden representation $\boldsymbol{Z}_0 \in \mathbb{R}^{D\times P}$ through an affine embedding and the addition of a positional encoding (PE) $\boldsymbol{P} \in \mathbb{R}^{D\times P}$. Formally, the pre-processing operation is defined as

$$\boldsymbol{Z}_0 = \mathcal{P}(\boldsymbol{x}) = (\boldsymbol{W}_E\boldsymbol{x} + \boldsymbol{b}_E)\boldsymbol{e}_1^\top + \boldsymbol{P},$$

where $\boldsymbol{W}_E \in \mathbb{R}^{D\times d}$ is the embedding weight matrix, $\boldsymbol{b}_E \in \mathbb{R}^D$ is the bias vector, and $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^\top \in \mathbb{R}^P$ is the canonical basis vector that assigns the embedded input exclusively to the first token position. Here, $P \geq 1$ denotes the total sequence length, implying that the remaining $P-1$ columns of $\boldsymbol{Z}_0$ serve as appended padding or latent tokens. We use sinusoidal positional encodings in our constructions, which aligns with the Transformer architectures in practical applications [44].

Consider the i-th encoder block with input $\boldsymbol{Z}_{i-1} = [\boldsymbol{z}_{i-1}^1, \ldots, \boldsymbol{z}_{i-1}^P] \in \mathbb{R}^{D\times P}$. Within the MHA layer $\mathcal{A}_i : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with H_i heads, each head includes a query matrix $\boldsymbol{Q}_i^h \in \mathbb{R}^{d_k\times D}$, a key matrix $\boldsymbol{K}_i^h \in \mathbb{R}^{d_k\times D}$, and a value matrix $\boldsymbol{V}_i^h \in \mathbb{R}^{d_v\times D}$, where d_k is the uniform query/key dimension and d_v is the uniform value dimension. For $h \in [H_i]$, the output of the h-th attention head $\mathrm{head}_i^h \in \mathbb{R}^{d_v\times P}$ is computed as

$$\mathrm{head}_i^h = \boldsymbol{V}_i^h\boldsymbol{Z}_{i-1}\boldsymbol{A}_i^h,$$

where $\boldsymbol{A}_i^h \in \mathbb{R}^{P\times P}$ is the attention probability matrix. Its j-th column is defined by

$$(\boldsymbol{A}_i^h)_{:,j} := \mathrm{Softmax}\left(\left(\boldsymbol{Z}_{i-1}^\top {\boldsymbol{K}_i^h}^\top \boldsymbol{Q}_i^h\boldsymbol{Z}_{i-1}\right)_{:,j}\right), \quad j \in [P],$$

with the $\mathrm{Softmax} : \mathbb{R}^P \to \mathbb{R}^P$ operator applied component-wise as

$$\mathrm{Softmax}(\boldsymbol{x}) = \left[\frac{e^{x_1}}{\sum_{l=1}^P e^{x_l}}, \ldots, \frac{e^{x_P}}{\sum_{l=1}^P e^{x_l}}\right]^\top.$$

To preserve the distinct representations from different heads, the final output of the MHA layer $\widehat{\boldsymbol{Z}}_i \in \mathbb{R}^{D\times P}$ is generated by concatenating the outputs of all H_i heads along the feature dimension, followed by a linear projection

$$\widehat{\boldsymbol{Z}}_i = \boldsymbol{W}_i^O \begin{bmatrix} \mathrm{head}_i^1 \\ \vdots \\ \mathrm{head}_i^{H_i} \end{bmatrix},$$

where $\boldsymbol{W}_i^O \in \mathbb{R}^{D\times(H_i\cdot d_v)}$ is the output projection matrix.

The output $\widehat{\boldsymbol{Z}}_i = [\widehat{\boldsymbol{z}}_i^1, \ldots, \widehat{\boldsymbol{z}}_i^P] \in \mathbb{R}^{D\times P}$ is subsequently fed into a point-wise FFN. The FFN layer $\mathcal{F}_i : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with weight matrices $\boldsymbol{W}_i^1 \in \mathbb{R}^{d_{\mathrm{ff}}\times D}$, $\boldsymbol{W}_i^2 \in \mathbb{R}^{D\times d_{\mathrm{ff}}}$ and biases $\boldsymbol{b}_i^1 \in \mathbb{R}^{d_{\mathrm{ff}}}$, $\boldsymbol{b}_i^2 \in \mathbb{R}^D$ computes

$$\boldsymbol{z}_i^j = \boldsymbol{W}_i^2 \sigma\left(\boldsymbol{W}_i^1 \widehat{\boldsymbol{z}}_i^j + \boldsymbol{b}_i^1\right) + \boldsymbol{b}_i^2, \quad j \in [P], \tag{2}$$

where $\sigma(\cdot) = \max(0, \cdot)$ is the ReLU activation function applied component-wise.

Finally, for the Transformer model with L encoder blocks, its output is

$$\mathcal{T}_L(\boldsymbol{x}) = \mathcal{F}_L \circ \mathcal{A}_L \circ \cdots \circ \mathcal{F}_1 \circ \mathcal{A}_1 \circ \mathcal{P}(\boldsymbol{x}) \in \mathbb{R}^{D\times P}.$$

The scalar output is then obtained by reading out the relevant components of $\mathcal{T}_L(\boldsymbol{x})$, via a linear affine mapping $\boldsymbol{c}_{L+1}^\top \mathrm{vec}(\mathcal{T}_L(\boldsymbol{x}))$. With the complete architecture established, we now define the function class expressed by such Transformer networks.

Definition 1 (Transformer Network Class)**.** *For depth $L \in \mathbb{N}$, embedding dimension $D \in \mathbb{N}$, sequence length $P \in \mathbb{N}$, layer configurations $\{H^\ell\}_{\ell=1}^L, \{d_k^\ell\}_{\ell=1}^L, \{d_v^\ell\}_{\ell=1}^L, \{d_{\mathrm{ff}}^\ell\}_{\ell=1}^L \subset \mathbb{N}$, and parameter magnitude bound $M > 0$, we define the class of Transformer networks as*

$$\mathcal{T}\left(L, D, P, \{H^\ell\}_{\ell=1}^L, \{d_k^\ell\}_{\ell=1}^L, \{d_v^\ell\}_{\ell=1}^L, \{d_{\mathrm{ff}}^\ell\}_{\ell=1}^L, M\right)$$
$$= \left\{ f_\theta \;\middle|\; \begin{array}{l} f_\theta(\boldsymbol{x}) = \boldsymbol{c}_{L+1}^\top \mathrm{vec}(\mathcal{T}_L(\boldsymbol{x})) \text{ is an } L\text{-block Transformer with embedding dim } D, \\ \text{sequence length } P, \text{parameter bound } \|\theta\|_\infty \le M, \text{ and for each block } \ell \in [L]: \\ \text{there is } H^\ell \text{heads, query/key dim is } d_k^\ell, \text{ value dim is } d_v^\ell, \text{ FFN hidden width is } d_{\mathrm{ff}}^\ell \end{array} \right\}.$$

3 Learning on Euclidean domains

In this section, we establish the theoretical capabilities of the Transformer architecture. We focus our analysis on compact Euclidean domains and introduce a novel Softmax POU framework. In Subsection 3.1, we constructively prove uniform approximation rates for Hölder continuous functions. Building upon these approximation results, we further establish the corresponding statistical generalization bounds in Subsection 3.2.

3.1 Approximation of Hölder functions on Euclidean domains

A foundational ingredient in our construction is the Softmax POU. Unlike deep ReLU networks that rely on locally compactly supported bump functions as POU [46, 47, 33], we introduce a globally supported "soft" POU by Softmax POU. This idea shares connections with classical non-parametric techniques, such as Shepard's method for spatial interpolation [35] and normalized Radial Basis Function (RBF) networks [28]. By utilizing a scaling parameter M_g to control the exponential decay outside local neighborhoods, this scheme naturally establishes a local-to-global approximation paradigm. We formalize this mathematical construction in the following lemma, which is proved in Appendix A.1.

Lemma 1 (Softmax POU Approximation on Euclidean Space)**.** *Let $d \in \mathbb{N}$, $\alpha \in (0, 1]$, and $g : [0,1]^d \to \mathbb{R}$ be an α-Hölder continuous function with constant C_H. For any target accuracy $\varepsilon \in (0, \frac{1}{e}]$, let the covering radius be $r_g = (\varepsilon/(4C_H))^{1/\alpha}$. There exists a finite set of C_g centers $\{\boldsymbol{c}_i\}_{i=1}^{C_g} \subset [0,1]^d$ forming an r_g-covering of the domain (i.e., $[0,1]^d \subseteq \bigcup_{i=1}^{C_g} \bar{\mathcal{B}}(\boldsymbol{c}_i, r_g)$) with $C_g \le \left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d \varepsilon^{-\frac{d}{\alpha}}$. By setting the scaling parameter*

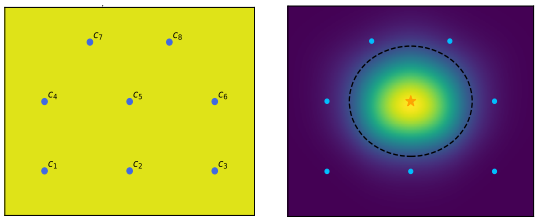
$$M_g = \frac{(4C_H)^{2/\alpha}}{3}\left(\log\left(4\|g\|_\infty \left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d\right) + \frac{d+\alpha}{\alpha}\right)\varepsilon^{-\frac{2}{\alpha}} \log\frac{1}{\varepsilon},$$

the Softmax POU approximation $\widehat{g}(\boldsymbol{x}) = \sum_{i=1}^{C_g} \beta_i(\boldsymbol{x}) g(\boldsymbol{c}_i)$, *with*

$$\beta_i(\boldsymbol{x}) = \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g} \exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2)\right)} = \frac{\exp\left(2M_g \langle \boldsymbol{x}, \boldsymbol{c}_i \rangle - M_g \|\boldsymbol{c}_i\|_2^2\right)}{\sum_{k=1}^{C_g} \exp\left(2M_g \langle \boldsymbol{x}, \boldsymbol{c}_k \rangle - M_g \|\boldsymbol{c}_k\|_2^2\right)}, \quad (3)$$

achieves the uniform error bound $\sup_{\boldsymbol{x} \in [0,1]^d} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \varepsilon$.

Lemma 1 introduces our new approximation framework. The centers $\{\boldsymbol{c}_i\}_{i=1}^{C_g}$ and the corresponding covering balls $\bar{\mathcal{B}}(\boldsymbol{c}_i, r_g)$ induce localized soft regions through the weights $\beta_i(\boldsymbol{x})$, which is illustrated in Figure 1. The weight functions $\{\beta_i(\boldsymbol{x})\}_{i=1}^{C_g}$ form a partition of unity over the domain, and the scaling parameter M_g controls the exponential decay of $\beta_i(\boldsymbol{x})$.



(a) Centers (b) $\beta_5(\boldsymbol{x})$

Figure 1: Geometric illustration of the softmax partition of unity. The centers $\{\boldsymbol{c}_i\}_{i=1}^{C_g}$ and covering balls $\bar{\mathcal{B}}(\boldsymbol{c}_i, r_g)$ induce the localized soft weights $\beta_i(\boldsymbol{x})$.

The Softmax POU approximation scheme offers two primary advantages: (i) the algebraic cancellation of the input-dependent quadratic term $\|\boldsymbol{x}\|_2^2$ eliminates the geometric quadratic form, reducing the relative distance computation to purely affine transformations; and (ii) it perfectly aligns with the native Transformer architecture, enabling mesh-free spatial localization that is dynamically achieved through standard softmax dot-product attention mechanism. The following theorem quantifies the approximation power of Transformers using the Softmax POU approximation framework.

Theorem 1 (Transformer Approximation on Euclidean Space)**.** *Let* $d \in \mathbb{N}$, $\alpha \in (0, 1]$, *and* $g : [0,1]^d \to \mathbb{R}$ *be* α*-Hölder continuous with constant* C_H *and* $\|g\|_\infty \leq B$. *For any* $\varepsilon \in (0, 1/e]$, *there exists a Transformer network*

$$\widehat{g}_T \in \mathcal{T}\Big(L, D, P, \{H^\ell\}_{\ell=1}^L, \{d_k^\ell\}_{\ell=1}^L, \{d_v^\ell\}_{\ell=1}^L, \{d_{ff}^\ell\}_{\ell=1}^L, M_{max}\Big)$$

satisfying

$$\sup_{\boldsymbol{x} \in [0,1]^d} |\widehat{g}_T(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \varepsilon$$

with the following parameters

- $L = 2$, $D = d + 4$, *and* $P \leq C_P \varepsilon^{-d/\alpha}$ *where* $C_P = d^{d/2}(8C_H)^{d/\alpha}$.
- *For* $\ell = 1$: $H^1 = P + 2$, $d_k^1 = d_v^1 = 2$, *and* $d_{ff}^1 = 2d + 8$.
- *For* $\ell = 2$: $H^2 = 1$, $d_k^2 = d_v^2 = 1$, *and* $d_{ff}^2 = 2d + 8$.

The parameter magnitude M_{max} *and total dense parameters* $\mathcal{N}_{total}$ *satisfy*

$$M_{max} \leq \widetilde{C}_{mag} \varepsilon^{-2d/\alpha} \log \frac{1}{\varepsilon}, \qquad \mathcal{N}_{total} \leq C_N \varepsilon^{-d/\alpha},$$

where $C_N = 10(d+4)C_P + 9d^2 + 95d + 236$, *and* $\widetilde{C}_{mag} = \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}, 2\left(1 + \frac{1}{2eB}\right) C_B\right\}$, *with* $C_B = \max\{3dC_M, B, 1\}$, $C_{mag} = \frac{C_P^2 C_{log}}{8}$, $C_{log} = |\log(4C_P B(1 + 6dC_M))| + \frac{d+2+\alpha}{\alpha} + 2$, *and* $C_M = \frac{(8C_H)^{2/\alpha}}{3}\left(\log\left(4Bd^{d/2}(4C_H)^{d/\alpha}\right) + \frac{d+\alpha}{\alpha}\right)$.

Theorem 1 places representational capacity in the sequence dimension P while retaining two dense encoder blocks and single-hidden-layer point-wise FFNs. Native dot-product softmax attention supplies spatial localization. Sinusoidal positional encodings on a uniform angular grid compute relative positions through phase differences; their cyclic symmetries preserve token identities across layers.

Proof Sketch of Theorem 1. The detailed proof of Theorem 1 is deferred to Appendix B.1. We construct a 2-block Transformer to execute the Softmax POU approximation scheme by leveraging the sequence dimension P as an index set to parallelize the localized expert approximations. The procedure is organized into four key stages:

Pre-processing (see Lemma 4): The input $\boldsymbol{x} \in [0,1]^d$ is mapped to an initial sequence matrix $\boldsymbol{Z}_0 \in \mathbb{R}^{D\times P}$. This layer embeds $\boldsymbol{x}$ and an indicator exclusively into the first token, zero-pads the remaining sequence, and appends sinusoidal positional encodings across all P tokens

$$(\boldsymbol{Z}_0)_{:,j} = \left[\delta_{j,1}\boldsymbol{x}^\top, \quad \delta_{j,1}, \quad 0, \quad \sin(\theta_j), \quad \cos(\theta_j)\right]^\top.$$

First MHA (Local Feature Extraction, see Lemma 5): Utilizing $H^1 = P + 2$ heads, this layer computes parallel affine transformations to extract local spatial features and approximation expert features. The output $\widehat{\boldsymbol{Z}}_1 = \mathcal{A}_1(\boldsymbol{Z}_0)$ takes the column-wise form

$$(\widehat{\boldsymbol{Z}}_1)_{:,j} = \left[\widetilde{T}_j(\boldsymbol{x}), \quad \widetilde{g}(\boldsymbol{c}_j), \quad \boldsymbol{0}_{d-1}^\top, \quad \widetilde{I}_j, \quad \lambda(M)\sin(\theta_j), \quad \lambda(M)\cos(\theta_j)\right]^\top,$$

where $\widetilde{T}_j(\boldsymbol{x}) \approx 2M_g\boldsymbol{c}_j^\top\boldsymbol{x} - M_g\|\boldsymbol{c}_j\|_2^2$ captures the approximated local spatial feature, $\widetilde{g}(\boldsymbol{c}_j) \approx g(\boldsymbol{c}_j)$ represents the approximated local expert feature for each center $\boldsymbol{c}_j$, and $\widetilde{I}_j \approx \delta_{j,1}$ serves as the approximated structural indicator for the first token. Crucially, in our construction, the exact angular identity θ_j of the sinusoidal positional encodings remains invariant up to a uniform scalar scaling $\lambda(M)$.

First FFN (Non-linear Restoration, see Lemma 6): By exploiting the hard-thresholding gating property of the ReLU activation, $\boldsymbol{Z}_1 = \mathcal{F}_1(\widehat{\boldsymbol{Z}}_1)$ functions as a non-linear filter to eliminate the small error introduced by the softmax operation, thereby exactly recovering the canonical form and original sinusoidal positional encodings

$$(\boldsymbol{Z}_1)_{:,j} = \left[\widetilde{T}_j(\boldsymbol{x}), \quad \widetilde{g}(\boldsymbol{c}_j), \quad \boldsymbol{0}_{d-1}^\top, \quad \delta_{j,1}, \quad \sin(\theta_j), \quad \cos(\theta_j)\right]^\top.$$

Second MHA (Global Aggregation, see Lemma 7): A single head ($H^2 = 1$) extracts the indicator $\delta_{k,1}$ via the Query, the local spatial features $\widetilde{T}_j(\boldsymbol{x})$ via the Key, and the local expert features $\widetilde{g}(\boldsymbol{c}_j)$ via the Value. The softmax activation then directly computes the normalized POU weights $\widetilde{\beta}_j(\boldsymbol{x}) = \exp(\widetilde{T}_j(\boldsymbol{x}))/\sum_{l=1}^P \exp(\widetilde{T}_l(\boldsymbol{x}))$, aggregating the local expert predictions into a unified global approximation $\widehat{g}_T(\boldsymbol{x})$ and storing the result in the first column of $\widehat{\boldsymbol{Z}}_2 = \mathcal{A}_2(\boldsymbol{Z}_1)$, that is

$$(\widehat{\boldsymbol{Z}}_2)_{:,1} = \left[\widehat{g}_T(\boldsymbol{x}), \quad \boldsymbol{0}_{D-1}^\top\right]^\top, \quad \text{where} \quad \widehat{g}_T(\boldsymbol{x}) = \sum_{j=1}^P \widetilde{\beta}_j(\boldsymbol{x})\widetilde{g}(\boldsymbol{c}_j).$$

Second FFN & Readout: Finally, $\mathcal{F}_2$ operates as an exact identity mapping $\boldsymbol{Z}_2 = \widehat{\boldsymbol{Z}}_2$. A linear readout vector $\boldsymbol{c}^\top$ explicitly extracts the final approximation $\widehat{g}_T(\boldsymbol{x})$ from the hidden state. □

3.2 Generalization analysis

In this section, we analyze the generalization capability of the empirical risk minimization (ERM) algorithm over the bounded hypothesis space constructed based on our Transformer architecture. We utilize the covering number to measure the capacity of the hypothesis space, which subsequently provides insights into the learning performance of the ERM algorithm.

We follow the classical learning framework for regression [7]. Suppose that a data sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X}\times\mathcal{Y}$ is independently drawn from a true unknown Borel probability measure ρ on $\mathcal{X}\times\mathcal{Y}$, with $\mathcal{Y} \subseteq [-B, B]$ for some $B > 0$. The target function for learning is the regression function $f_\rho : \mathcal{X} \to \mathbb{R}$ defined by $f_\rho(\boldsymbol{x}) = \int_{\mathcal{Y}} y d\rho(y|\boldsymbol{x})$, which minimizes the expected generalization error

$$\mathcal{E}(f) := \int_{\mathcal{X}\times\mathcal{Y}} (f(\boldsymbol{x}) - y)^2 d\rho.$$

We denote by $\rho_{\mathcal{X}}$ the marginal distribution of ρ on $\mathcal{X}$, and by $(L^2_{\rho_{\mathcal{X}}}, \|\cdot\|_\rho)$ the Hilbert space of square-integrable functions with respect to $\rho_{\mathcal{X}}$, where $\|f\|_\rho^2 := \int_{\mathcal{X}} |f(\boldsymbol{x})|^2 \, d\rho_{\mathcal{X}}(\boldsymbol{x})$. For every $f \in L^2_{\rho_{\mathcal{X}}}$, the squared-loss excess risk satisfies $\mathcal{E}(f) - \mathcal{E}(f_\rho) = \|f - f_\rho\|_\rho^2$.

We consider the ERM algorithm over the Transformer hypothesis space $\mathcal{T}$ defined in Theorem 1. Based on this hypothesis space, the empirical target function is defined as

$$f_{\mathcal{S}} := \arg\min_{f\in\mathcal{T}} \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i)-y_i)^2.$$

Since $\mathcal{Y}\subseteq[-B,B]$, we project the output function onto the interval $[-B,B]$ and define the truncated empirical target function as $\pi_B f_{\mathcal{S}}$, where $\pi_B(t)=\max\{-B,\min\{t,B\}\}=-B+\mathrm{ReLU}(t+B)-\mathrm{ReLU}(t-B)$. Thus, clipping is implemented exactly by one hidden ReLU layer with two units and an affine readout.

For target functions defined on Euclidean domains, based on the approximation error bound in Theorem 1 and the covering number bound in Lemma 9 in Appendix C.1, we obtain the following near minimax-optimal convergence rate of the Transformer estimator $f_{\mathcal{S}}$.

Theorem 2. *Suppose that the regression function $f_\rho : [0,1]^d \to \mathbb{R}$ is an α-Hölder continuous function with constant C_H and $\|f_\rho\|_\infty \le B$. Consider the truncated empirical target function $\pi_B f_{\mathcal{S}}$ over the hypothesis space $\mathcal{T}$ defined in Theorem 1, for $0<\varepsilon\le 1/e$, we have*

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}}-f_\rho\|_\rho^2\right] \le C_4 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\}.$$

Moreover, by choosing $\varepsilon = n^{-\frac{\alpha}{2\alpha+d}}$, we obtain

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}}-f_\rho\|_\rho^2\right] \le C_4 n^{-\frac{2\alpha}{2\alpha+d}}\log n,$$

where C_4 is a positive constant depending on d,α,B, and C_H specified in (14).

Proof Sketch of Theorem 2. The detailed proof is deferred to Appendix C. The result follows from optimizing the bias-variance trade-off established via an oracle inequality (Lemma 10). First, we bound the approximation error (bias) by ε utilizing the Transformer construction from Theorem 1. Second, we control the estimation error (variance) by bounding the capacity of the hypothesis space $\mathcal{T}$ as detailed in Appendix C.1. Specifically, Lemma 9 provides the covering number bound

$$\log\mathcal{N}(\eta,\mathcal{T},\|\cdot\|_\infty) \le \mathcal{N}_{total}\log\left(\frac{1224256P^4D^{22}M_{max}^{26}}{\eta}\right).$$

Finally, balancing the approximation and estimation errors and choosing the optimal network accuracy $\varepsilon = n^{-\frac{\alpha}{2\alpha+d}}$ yields the desired convergence rate as detailed in Appendix C.2. □

4 Adaptivity to Riemannian manifolds

In this section, we assume that the input domain $\mathcal{M}\subseteq[0,1]^{\bar{d}}$ is a compact and connected d-dimensional Riemannian manifold with strictly positive reach $\tau>0$ (see Definition 2 in Appendix A.2). Our goal is to investigate the theoretical capabilities of Transformers for learning α-Hölder continuous functions defined on $\mathcal{M}$ (see Definition 3 in Appendix A.2). More details about manifolds can be found in [43, 23].

Our Softmax POU approximation scheme naturally adapts to complex geometries by operating directly on the ambient Euclidean metric. Because the ReLU activation is fundamentally piecewise linear, locally supported ReLU-based POUs are geometrically constrained to partition space using hyperplanes, such as explicit Cartesian grids or simplicial triangulations. In contrast, the Softmax POU is inherently mesh-free. By exploiting the local metric equivalence between intrinsic geodesic and ambient Euclidean distances, it achieves spatial localization purely through affine transformations, avoiding the need for explicit surface parameterizations or complex non-linear geodesic computations. This exact formulation perfectly aligns with the native dot-product attention mechanism in Transformers. Building upon this geometric intuition, the following theorem formally establishes how Transformers natively execute this local-to-global approximation scheme to bypass the geometric complexities of compact Riemannian manifolds.

Theorem 3 (Transformer Approximation on Manifolds). *Let $\mathcal{M} \subseteq [0,1]^{\bar{d}}$ be a d-dimensional compact and connected Riemannian manifold with reach $\tau > 0$, and let $g : \mathcal{M} \to \mathbb{R}$ be an α-Hölder continuous function with constant C_H and $\|g\|_\infty \leq B$. For any ε satisfying $0 < \varepsilon \leq \min\left\{\frac{1}{e}, 16C_H\left(\frac{\tau}{4}\right)^\alpha\right\}$, there exists a Transformer network*

$$\widehat{g}_T \in \mathcal{T}\Big(L, \bar{d}+4, P, \{H^\ell\}_{\ell=1}^L, \{d_k^\ell\}_{\ell=1}^L, \{d_v^\ell\}_{\ell=1}^L, \{d_{ff}^\ell\}_{\ell=1}^L, M_{max}\Big)$$

satisfying

$$\sup_{\boldsymbol{x}\in\mathcal{M}} |\widehat{g}_T(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \varepsilon$$

with the following parameters:

- *$L = 2$, embedding dimension $D = \bar{d}+4$, and $P \leq C_P \varepsilon^{-d/\alpha}$ where $C_P = C_{\mathcal{M}}(16C_H)^{d/\alpha}$.*
- *For $\ell = 1$: $H^1 = P+2$, $d_k^1 = d_v^1 = 2$, and $d_{ff}^1 = 2\bar{d}+8$.*
- *For $\ell = 2$: $H^2 = 1$, $d_k^2 = d_v^2 = 1$, and $d_{ff}^2 = 2\bar{d}+8$.*

The parameter magnitude M_{max} and total dense parameters $\mathcal{N}_{total}$ satisfy

$$M_{max} \leq \widetilde{C}_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, \qquad \mathcal{N}_{total} \leq C_N\varepsilon^{-d/\alpha},$$

where $C_N = 10(\bar{d}+4)C_P + 9\bar{d}^2 + 95\bar{d} + 236$, $\widetilde{C}_{mag} = \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}, 2\left(1+\frac{1}{2eB}\right)C_B\right\}$, with $C_B = \max\{3\bar{d}C_M, B, 1\}$, $C_{mag} = \frac{C_P^2 C_{log}}{8}$, $C_{log} = \left|\log(2C_P B(1+6\bar{d}C_M))\right| + \frac{d+2+\alpha}{\alpha} + 2$, $C_M = \frac{(16C_H)^{2/\alpha}}{3}\left(\log(4BC_P) + \frac{d+\alpha}{\alpha}\right)$, $C_{\mathcal{M}} = 3^d Vol(\mathcal{M})d^{d/2}$, and $Vol(\mathcal{M})$ is the volume of $\mathcal{M}$.

The rigorous establishment of this approximation guarantee relies on several key geometric tools detailed in Appendix A.2. Specifically, we leverage the local metric equivalence between the geodesic and Euclidean distances governed by the reach τ, alongside the intrinsic covering number bounds of manifolds (see Lemma 2). These geometric properties theoretically validate the extension of our Softmax POU scheme to the manifold setting (see Lemma 3). The detailed proof of Theorem 3 is deferred to Appendix B.2. The architectural construction follows a similar procedure to the proof of Theorem 1, but adapts the scaling factor and sequence length to the covering properties of $\mathcal{M}$. Based on this approximation theorem, we further establish the statistical convergence rate of the Transformer estimator $f_{\mathcal{S}}$ for target functions defined on manifolds.

Theorem 4. *Suppose that the regression function $f_\rho : \mathcal{M} \to \mathbb{R}$ is an α-Hölder continuous function with constant C_H and $\|f_\rho\|_\infty \leq B$, where $\mathcal{M} \subseteq [0,1]^{\bar{d}}$ is a compact and connected d-dimensional Riemannian manifold with reach $\tau > 0$. Consider the truncated empirical target function $\pi_B f_{\mathcal{S}}$ over the hypothesis space $\mathcal{T}$ defined in Theorem 3, for $0 < \varepsilon \leq \min\left\{\frac{1}{e}, 16C_H\left(\frac{\tau}{4}\right)^\alpha\right\}$, we have*

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] \leq \widetilde{C}_4 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\}.$$

Moreover, by choosing $\varepsilon = n^{-\frac{\alpha}{2\alpha+d}}$, we obtain

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] \leq \widetilde{C}_4 n^{-\frac{2\alpha}{2\alpha+d}}\log n,$$

where $\widetilde{C}_4$ is a positive constant depending on $\bar{d}, d, \alpha, B, C_H$, and $Vol(\mathcal{M})$ specified in (15).

The detailed proof of Theorem 4 is deferred to Appendix C. Similar to the Euclidean setting, Theorem 4 is derived by utilizing the bias-variance tradeoff and the covering number bound of the Transformer hypothesis space. Crucially, notice that the the exponent of the final learning rate depends exclusively on the intrinsic dimension d rather than the ambient dimension $\bar{d}$. This demonstrates that a shallow, wide, and dense Transformer hypothesis class can effectively avoid the curse of dimensionality when learning from data intrinsically supported on low-dimensional manifolds.

5 Numerical experiment

We examine whether Transformer training produces spatially localized attention on a compact manifold. We consider noiseless regression $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^8$ on the unit circle with Cartesian inputs $\boldsymbol{x}(u) = (\cos(2\pi u - \pi), \sin(2\pi u - \pi))^\top$, $u \in [0, 1)$. The eight target functions $g_1, \ldots, g_8$ are generated independently as continuous, periodic, piecewise-linear functions of u, each with eight segments (Appendix D.1, (16)). We jointly fit the eight targets using 1024 inputs sampled uniformly from the circle. Evaluation uses 4096 equally spaced circle points. The Transformer we use has 128 tokens of dimension 16 (Appendix D.1, (17)). An affine map inserts $\boldsymbol{x}$ into token 0; the other 127 tokens have learned embeddings with no prescribed spatial locations. The first block combines two-head self-attention and a ReLU FFN with residual connections, producing $\boldsymbol{Z}^{(1)}(\boldsymbol{x}) = [\boldsymbol{z}_0(\boldsymbol{x}), \ldots, \boldsymbol{z}_{127}(\boldsymbol{x})]$. To isolate attention-weighted aggregation, the second block uses one-head attention from query token 0 to sources $\mathcal{S} = \{1, \ldots, 127\}$, with an identity FFN and no residual connection:

$$
\begin{aligned}
a_j(\boldsymbol{x}) &= \frac{\exp\{(\boldsymbol{Q}_2 \boldsymbol{z}_0(\boldsymbol{x}))^\top \boldsymbol{K}_2 \boldsymbol{z}_j(\boldsymbol{x})\}}{\sum_{k \in \mathcal{S}} \exp\{(\boldsymbol{Q}_2 \boldsymbol{z}_0(\boldsymbol{x}))^\top \boldsymbol{K}_2 \boldsymbol{z}_k(\boldsymbol{x})\}}, \\
\boldsymbol{f}(\boldsymbol{x}) &= \sum_{j \in \mathcal{S}} a_j(\boldsymbol{x}) \boldsymbol{e}_j(\boldsymbol{x}), \qquad \boldsymbol{e}_j(\boldsymbol{x}) = \boldsymbol{W}_V^{\text{eff}} \boldsymbol{z}_j(\boldsymbol{x}) + \boldsymbol{b}_V \in \mathbb{R}^8.
\end{aligned}
\tag{4}
$$

We minimize the squared loss (18) using 14000 Adam updates (initial learning rate 0.003, batch size 256), followed by up to 400 full-batch L-BFGS iterations. Learning-rate scheduling and checkpoint selection are specified in Appendix D.1. The network achieves normalized mean squared error (NMSE; (19)) of 9.07×10^{-4}.

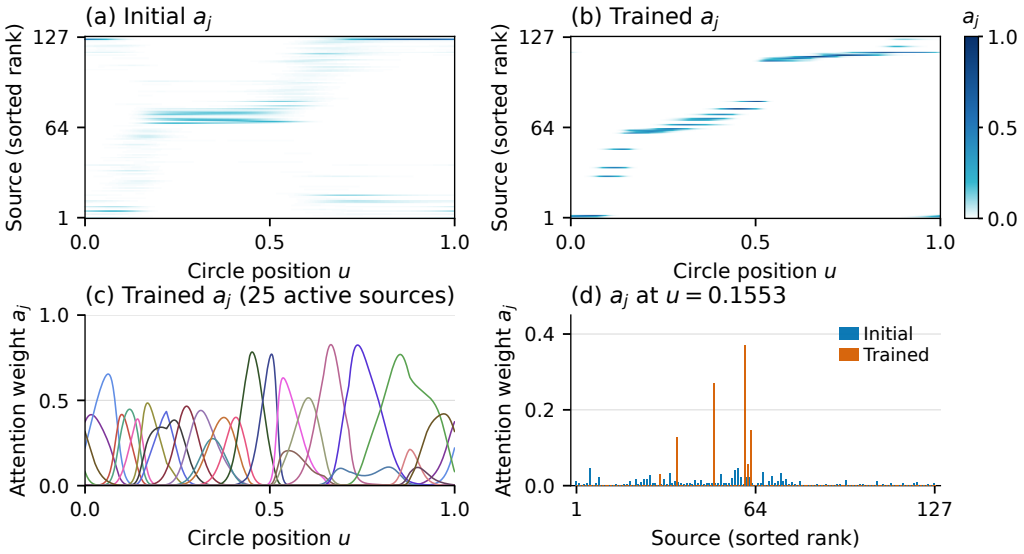


Figure 2: Second-layer softmax attention weights $a_j(\boldsymbol{x}(u))$. Panels (a,b,d) share the source order sorted by trained calibration peaks. Panel (c) shows all 25 active sources whose mean trained attention weight over calibration inputs meets the threshold in (20). Selection details: Appendix D.2.

Figure 2(a,b) illustrates how $a_j(\boldsymbol{x}(u))$ evolve during training. Initially, attention responses are relatively diffuse across the circle. After training, different sources develop more spatially concentrated responses, becoming prominent in different local regions. Panel (c) further reveals overlapping localized bumps from the 25 active sources, many of which exhibit Gaussian-like weights with rapid decay away from their peaks. These patterns resemble the Softmax POU weights in (1), where localized weights emphasize nearby regions while maintaining smooth transitions through their overlap. Panel (d) illustrates this input-dependent weighting at $u = 0.1553$, where the trained attention concentrates most of its weights on a few sources, in contrast to the more diffuse initial distribution. Together, these observations suggest that spatially localized, Gaussian-like attention weights can emerge through training without prescribed centers or locality penalties, exhibiting a local-to-global weighting behavior qualitatively aligned with our Softmax POU construction.

Table 1: Comparison of the cited Transformer constructions and our work. The reported depth includes the hidden depth in point-wise FFN layers.

References	Softmax Attention	Sinusoidal PE	Depth
Gurevych et al. [17]	✗	✗	$\mathcal{O}(1)$
Takakura and Suzuki [42]	✓	✓	$\mathcal{O}(\text{poly}(\log(1/\varepsilon)))$
Havrilla and Liao [18]	✗	✓	$\mathcal{O}(\log(1/\varepsilon))$
Shi et al. [37]	✓	✗	$\mathcal{O}(\log(1/\varepsilon))$
Jiao et al. [20]	✓	✗	$\mathcal{O}(\log(1/\varepsilon))$
Ours (This work)	✓	✓	2

6 Related work and discussion

Learning Theory of Transformers. Recent studies have investigated the quantitative approximation and estimation capabilities of Transformers [17, 42, 19, 18, 37, 20]. Specifically, they show that Transformers can mitigate the curse of dimensionality when target functions exhibit hierarchical compositional structures [17, 37] or anisotropic smoothness in sequence-to-sequence tasks [42]. Further research has established Jackson-type rates and comparisons with RNNs [19], and utilized the Kolmogorov-Arnold theorem for Hölder function approximation [20]. Some existing constructions replace softmax with hardmax-based or ReLU attention [17, 18], while others omit sinusoidal PE [37, 20]. Even when softmax attention is retained, existing approximation constructions still rely primarily on point-wise ReLU FFN layers [42, 37]. In contrast, our framework leverages native softmax attention to implement a Softmax POU for spatial localization and the aggregation of local approximations, while preserving sinusoidal positional encodings. Table 1 summarizes these architectural differences.

Learning Theory on Manifolds. Neural-network approximation and statistical learning have been studied on manifolds and more general low-dimensional sets. Chen et al. [3, 4] construct sparse ReLU approximations of smooth manifold functions and establish least-squares regression guarantees, with intrinsic dimension governing the leading rate exponents. Nakada and Imaizumi [30] obtain constant-depth ReLU approximation and regression bounds for Hölder functions with covariates supported on sets of low Minkowski dimension. Cloninger and Klock [6] analyze ReLU approximation of functions composed with manifold projections or distances to low-complexity sets, allowing full-dimensional domains. Liu et al. [24] establish Besov-function approximation on manifolds using convolutional residual networks and derive excess-logistic-risk bounds for binary classification. Dahal et al. [8] prove Wasserstein-1 approximation and estimation guarantees for ReLU generators of manifold-supported distributions with densities bounded above and below. Liu et al. [25] study chart autoencoders trained on paired noisy and clean samples, proving reconstruction and denoising guarantees under bounded normal noise. Havrilla and Liao [18] derive approximation and regression bounds for Transformers with ReLU attention, relating model-size and sample-size scaling exponents to intrinsic dimension and Hölder regularity. Nagler and Langer [29] establish dense ReLU approximation on sets of low Minkowski dimension with flexible width–depth tradeoffs, and derive regression rates adapted to intrinsic dimension, smoothness, and compositional structure.

7 Conclusion

In this paper, we establish a novel local-to-global constructive approximation framework for Transformers based on a globally supported Softmax POU. Leveraging this framework, we provide comprehensive approximation and generalization guarantees for α-Hölder continuous functions on both compact Euclidean domains and Riemannian manifolds, where we restrict $\alpha \in (0, 1]$. Specifically, we prove that shallow, wide, and dense Transformers with softmax activation and sinusoidal positional encodings can attain near minimax-optimal convergence rates and overcome the curse of dimensionality. A compelling future direction is to explore the potential of our Softmax POU framework for analyzing the mechanisms of in-context learning [2, 9, 34]. While classic studies often interpret this phenomenon by viewing Transformers as implicit statisticians [1] or as executing internal learning algorithms [13], our constructive scheme offers a potentially distinct perspective to further investigate how Transformer architectures achieve adaptive approximations during inference.

Acknowledgments and Disclosure of Funding

Zhongjie Shi and Wenjing Liao acknowledge support from the National Science Foundation under grant NSF DMS 2145167 and the U.S. Department of Energy under grant DOE SC0024348. AI tools were used to create Figure 1, implement, run, and analyze the numerical experiments, generate the experimental plots (Figures 2–3), and polish the language of the manuscript.

References

[1] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in Neural Information Processing Systems*, 36:57125–57211, 2023.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[3] Minshuo Chen, Haoming Jiang, Wenjing Liao, and Tuo Zhao. Efficient approximation of deep relu networks for functions on low dimensional manifolds. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] Minshuo Chen, Haoming Jiang, Wenjing Liao, and Tuo Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks: Function approximation and statistical recovery. *Information and Inference: A Journal of the IMA*, 11(4):1203–1253, 2022.

[5] Charles K Chui, Shao-Bo Lin, and Ding-Xuan Zhou. Deep net tree structure for balance of capacity and approximation ability. *Frontiers in Applied Mathematics and Statistics*, 5:46, 2019.

[6] Alexander Cloninger and Timo Klock. A deep network construction that adapts to intrinsic dimensionality beyond the domain. *Neural Networks*, 141:404–419, 2021.

[7] Felipe Cucker and Ding Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.

[8] Biraj Dahal, Alexander Havrilla, Minshuo Chen, Tuo Zhao, and Wenjing Liao. On deep generative models for approximation and estimation of distributions on manifolds. *Advances in Neural Information Processing Systems*, 35:10615–10628, 2022.

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[11] Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

[12] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491, 1959.

[13] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[14] Christopher R Genovese, Marco Perone Pacifico, Isabella Verdinelli, Larry Wasserman, et al. Minimax manifold estimation. *Journal of Machine Learning Research*, 13:1263–1291, 2012.

[15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. Ieee, 2013.

[17] Iryna Gurevych, Michael Kohler, and Gözde Gül Şahin. On the rate of convergence of a classifier based on a transformer encoder. *IEEE Transactions on Information Theory*, 68(12):8139–8155, 2022.

[18] Alex Havrilla and Wenjing Liao. Understanding scaling laws with statistical and approximation theory for transformer neural networks on intrinsically low-dimensional data. *Advances in Neural Information Processing Systems*, 37:42162–42210, 2024.

[19] Haotian Jiang and Qianxiao Li. Approximation rate of the transformer architecture for sequence modeling. *Advances in Neural Information Processing Systems*, 37:68926–68955, 2024.

[20] Yuling Jiao, Yanming Lai, Yang Wang, and Bokai Yan. Transformers can overcome the curse of dimensionality: A theoretical study from an approximation perspective. *Journal of Machine Learning Research*, 27(50):1–34, 2026.

[21] Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[23] John M Lee. *Riemannian Manifolds: An Introduction to Curvature*. Springer Science & Business Media, 2006.

[24] Hao Liu, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. In *International Conference on Machine Learning*, pages 6770–6780. PMLR, 2021.

[25] Hao Liu, Alex Havrilla, Rongjie Lai, and Wenjing Liao. Deep nonparametric estimation of intrinsic data structures by chart autoencoders: Generalization error and robustness. *Applied and Computational Harmonic Analysis*, 68:101602, 2024.

[26] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[27] Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Theory of deep convolutional neural networks iii: Approximating radial functions. *Neural Networks*, 144:778–790, 2021.

[28] John Moody and Christian J Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[29] Thomas Nagler and Sophie Langer. Optimal neural network approximation of smooth compositional functions on sets with low intrinsic dimension. *arXiv preprint arXiv:2602.03539*, 2026.

[30] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174):1–38, 2020.

[31] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1):419–441, 2008.

[32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[33] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

[34] Zhaiming Shen, Alexander Hsu, Rongjie Lai, and Wenjing Liao. Understanding in-context learning on structured manifolds: Bridging attention to kernel methods. In *The Fourteenth International Conference on Learning Representations*, 2026.

[35] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524, 1968.

[36] Zhongjie Shi, Jun Fan, Linhao Song, Ding-Xuan Zhou, and Johan AK Suykens. Nonlinear functional regression by functional deep neural network with kernel embedding. *Journal of Machine Learning Research*, 26(284):1–49, 2025.

[37] Zhongjie Shi, Zhiying Fang, and Yuan Cao. Approximation and estimation capability of vision transformers for hierarchical compositional models. *Applied and Computational Harmonic Analysis*, page 101849, 2025.

[38] Zhongjie Shi, Fanghui Liu, Yuan Cao, and Johan AK Suykens. Can overfitted deep neural networks in adversarial training generalize? an approximation viewpoint. *SIAM Journal on Mathematics of Data Science*, 8(2):225–256, 2026.

[39] Zhongjie Shi, Zhan Yu, and Ding-Xuan Zhou. Learning theory of distribution regression with neural networks. *Constructive Approximation*, 62(1):61–104, 2025.

[40] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[42] Shokichi Takakura and Taiji Suzuki. Approximation and estimation ability of transformers for sequence-to-sequence functions with infinite dimensional input. In *International Conference on Machine Learning*, pages 33416–33447. PMLR, 2023.

[43] Loring W Tu. Manifolds. In *An Introduction to Manifolds*, pages 47–83. Springer, 2011.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[45] Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 12071–12083, 2022.

[46] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[47] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. In *Conference on Learning Theory*, pages 639–649. PMLR, 2018.

[48] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.

[49] Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O (n) connections are expressive enough: Universal approximability of sparse transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 13783–13794, 2020.

[50] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297, 2020.

[51] Yufeng Zhang, Boyi Liu, Qi Cai, Lingxiao Wang, and Zhaoran Wang. An analysis of attention via the lens of exchangeability and latent variable models. *arXiv preprint arXiv:2212.14852*, 2022.

Appendix

This appendix provides the complete mathematical proofs for our theoretical results. It is organized as follows: Section A proves the preliminary lemmas for the Softmax POU scheme and Transformer constructions; Section B establishes the main approximation guarantees for both Euclidean domains and Riemannian manifolds; Section C details the covering number bounds and the derivation of the generalization error rates; and Section D provides the numerical protocols and additional diagnostics.

A Proof of preliminary results

This section establishes the detailed proofs for the foundational mathematical tools used in our theoretical framework. We first prove the Softmax POU approximation schemes and then provide the constructive lemmas for individual Transformer layers.

A.1 Softmax POU scheme for Euclidean domains in Section 3

In this subsection, we present the proof of Lemma 1. This establishes the core mechanism of using a Softmax POU to achieve local-to-global approximation on compact Euclidean spaces.

Proof of Lemma 1. To construct the approximation, we first explicitly establish the finite covering and bound the required number of centers C_g. For any target accuracy $\varepsilon \in (0, 1/e]$, we set the covering radius as $r_g = (\varepsilon/(4C_H))^{1/\alpha}$. We partition the domain $[0,1]^d$ into a uniform grid of hypercubes. By strategically setting the side length of each hypercube to $\delta = 2r_g/\sqrt{d}$, the distance from its center to any vertex (i.e., the circumscribed radius) is exactly $\frac{1}{2}\sqrt{d\delta^2} = r_g$. This guarantees that a closed Euclidean ball $\bar{\mathcal{B}}(\boldsymbol{c}_k, r_g)$ centered at the midpoint $\boldsymbol{c}_k$ of each hypercube completely covers it, yielding a valid finite cover of $[0,1]^d$.

The number of grid points required along each dimension is exactly $\lceil 1/\delta \rceil$, resulting in a total of $C_g = \lceil 1/\delta \rceil^d$ centers. By the inequality that $\lceil x \rceil \le 2x$ when $x \ge 1$, we have

$$C_g = \left\lceil \frac{1}{\delta} \right\rceil^d = \left\lceil \frac{\sqrt{d}}{2r_g} \right\rceil^d \le \left(2 \cdot \frac{\sqrt{d}}{2r_g} \right)^d = \left(\frac{\sqrt{d}}{r_g} \right)^d = \left(\sqrt{d}(4C_H)^{1/\alpha} \right)^d \varepsilon^{-\frac{d}{\alpha}}.$$

With the centers $\{\boldsymbol{c}_i\}_{i=1}^{C_g}$ and the covering established, for any $\boldsymbol{x} \in [0,1]^d$, denote $i^*(\boldsymbol{x}) = \operatorname{argmin}_{i \in [C_g]} \|\boldsymbol{x} - \boldsymbol{c}_i\|_2$. Since the closed balls cover the domain, we are guaranteed that $\|\boldsymbol{x} - \boldsymbol{c}_{i^*}\|_2 \le r_g$.

Next, we bound the approximation error. Notice that $\beta_i(\boldsymbol{x}) = \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g} \exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_k\|_2^2)\right)}$, using the exact POU property that $\sum_{i=1}^{C_g} \beta_i(\boldsymbol{x}) = 1$, we decompose the error into near and far components at a radius of $2r_g$, that is

$$\begin{aligned} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| &= \left| \sum_{i=1}^{C_g} \beta_i(\boldsymbol{x}) \big(g(\boldsymbol{c}_i) - g(\boldsymbol{x}) \big) \right| \\ &\le \underbrace{\sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2 \le 2r_g} \beta_i(\boldsymbol{x}) |g(\boldsymbol{x}) - g(\boldsymbol{c}_i)|}_{(I)} + \underbrace{\sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2 > 2r_g} \beta_i(\boldsymbol{x}) |g(\boldsymbol{x}) - g(\boldsymbol{c}_i)|}_{(II)}. \end{aligned}$$

For the near-center terms (I), the α-Hölder continuity yields $|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \le C_H(2r_g)^\alpha \le 2C_H r_g^\alpha$ since $2^\alpha \le 2$ for $\alpha \in (0,1]$. By substituting our explicit choice of r_g, we obtain

$$(I) \le 2C_H r_g^\alpha \sum_i \beta_i(\boldsymbol{x}) \le 2C_H r_g^\alpha = 2C_H \left(\frac{\varepsilon}{4C_H} \right) = \frac{\varepsilon}{2}.$$

For the far-center tail terms (II), the condition $\|\boldsymbol{x}-\boldsymbol{c}_i\|_2 > 2r_g$ implies $r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_i\|_2^2 < -3r_g^2$. Since $\|\boldsymbol{x}-\boldsymbol{c}_{i^*}\|_2 \le r_g$, the denominator of $\beta_i(\boldsymbol{x})$ is bounded below

$$\sum_{k=1}^{C_g} \exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_k\|_2^2)\right) \ge \exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_{i^*}\|_2^2)\right) \ge \exp(0) = 1.$$

Applying the global bound $|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \le 2\|g\|_\infty$, the term (II) is bounded by

$$\begin{aligned}(II) &= \sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2>2r_g} \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_k\|_2^2)\right)} |g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \\ &\le \sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2>2r_g} \exp(-3M_g r_g^2)\cdot 2\|g\|_\infty \\ &\le 2C_g\|g\|_\infty \exp(-3M_g r_g^2).\end{aligned}$$

To guarantee $(II) \le \varepsilon/2$, the scaling parameter must satisfy $M_g \ge (3r_g^2)^{-1}\log(4C_g\|g\|_\infty/\varepsilon)$. Substituting the explicit formulas for r_g and C_g, we require

$$\begin{aligned}M_g &\ge \frac{(4C_H)^{2/\alpha}}{3}\varepsilon^{-\frac{2}{\alpha}} \log\left(4\|g\|_\infty\left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d \varepsilon^{-\frac{d+\alpha}{\alpha}}\right) \\ &= \frac{(4C_H)^{2/\alpha}}{3}\varepsilon^{-\frac{2}{\alpha}}\left(\log\left(4\|g\|_\infty\left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d\right) + \frac{d+\alpha}{\alpha}\log\frac{1}{\varepsilon}\right).\end{aligned}$$

Assuming $\varepsilon \le 1/e$ so that $\log(1/\varepsilon) \ge 1$, we can upper bound the bracketed sum by factoring out $\log(1/\varepsilon)$. Defining M_g as

$$M_g = \frac{(4C_H)^{2/\alpha}}{3}\left(\log\left(4\|g\|_\infty\left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d\right) + \frac{d+\alpha}{\alpha}\right)\varepsilon^{-\frac{2}{\alpha}}\log\frac{1}{\varepsilon}$$

satisfies this condition, ensuring $(II) \le \varepsilon/2$.

Combining (I) and (II) achieves the target uniform error

$$\sup_{\boldsymbol{x}\in[0,1]^d} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Finally, notice that the quadratic term $\|\boldsymbol{x}\|_2^2$ exactly cancels out in the Softmax expansion, we have

$$\begin{aligned}\beta_i(\boldsymbol{x}) &= \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x}\|_2^2)\right)\exp\left(2M_g\langle\boldsymbol{x},\boldsymbol{c}_i\rangle - M_g\|\boldsymbol{c}_i\|_2^2\right)}{\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2 - \|\boldsymbol{x}\|_2^2)\right)\exp\left(2M_g\langle\boldsymbol{x},\boldsymbol{c}_k\rangle - M_g\|\boldsymbol{c}_k\|_2^2\right)} \\ &= \frac{\exp\left(2M_g\langle\boldsymbol{x},\boldsymbol{c}_i\rangle - M_g\|\boldsymbol{c}_i\|_2^2\right)}{\sum_{k=1}^{C_g}\exp\left(2M_g\langle\boldsymbol{x},\boldsymbol{c}_k\rangle - M_g\|\boldsymbol{c}_k\|_2^2\right)},\end{aligned}$$

thus we complete the proof. □

A.2 Softmax POU scheme for manifolds in Section 4

Here, we extend our Softmax POU approximation scheme to compact Riemannian manifolds (See Lemma 3). We leverage the intrinsic geometric properties and local metric equivalence to properly bound the approximation errors. We first introduce some definitions and notations about Riemannian manifolds. Let $d(\mathbf{x},\mathcal{M}) := \inf_{\mathbf{v}\in\mathcal{M}}\|\mathbf{x}-\mathbf{v}\|_2$ denote the Euclidean distance from a point $\mathbf{x}\in\mathbb{R}^{\bar{d}}$ to a manifold $\mathcal{M}\subset\mathbb{R}^{\bar{d}}$. To characterize the curvature and the folding of $\mathcal{M}$ within the ambient space, we introduce the concept of reach.

Definition 2 (Reach [12, 31])**.** *The reach of $\mathcal{M}$ is defined as*

$$\tau = \inf_{\mathbf{v}\in\mathcal{M}}\inf_{\mathbf{x}\in G}\|\mathbf{x}-\mathbf{v}\|_2,$$

where $G = \left\{\mathbf{x}\in\mathbb{R}^{\bar{d}} : \exists \text{ distinct } \mathbf{p},\mathbf{q}\in\mathcal{M} \text{ such that } d(\mathbf{x},\mathcal{M}) = \|\mathbf{x}-\mathbf{p}\|_2 = \|\mathbf{x}-\mathbf{q}\|_2\right\}$ *is the medial axis of $\mathcal{M}$.*

We equip the compact and connected Riemannian manifold $\mathcal{M}$ with the intrinsic geodesic metric $d_{\mathcal{M}} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$, defined as

$$d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y}) := \inf \left\{ \int_0^1 \|\dot{\gamma}(t)\|_2 \, dt \;\middle|\; \gamma \in C^1([0,1], \mathcal{M}), \gamma(0) = \boldsymbol{x}, \gamma(1) = \boldsymbol{y} \right\}.$$

Based on this metric space $(\mathcal{M}, d_{\mathcal{M}})$, we define the Hölder continuous functions on the manifolds.

Definition 3 (Hölder Continuous Functions on Manifolds)**.** *Let $\alpha \in (0, 1]$, $(\mathcal{M}, d_{\mathcal{M}})$ be a compact and connected d-dimensional Riemannian manifold. A function $f : \mathcal{M} \to \mathbb{R}$ is α-Hölder continuous if there exists a constant $C_H \geq 0$ such that*

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq C_H \cdot [d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y})]^{\alpha}, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}.$$

Recall that the ϵ-covering number $\mathcal{N}(X, \epsilon, d)$ is the minimum number of open balls of radius ϵ needed to cover a metric space (X, d). Since the Euclidean chordal distance is naturally upper-bounded by the intrinsic geodesic distance (i.e., $\|\boldsymbol{x}-\boldsymbol{y}\|_2 \leq d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y})$), any valid ϵ-covering under $d_{\mathcal{M}}$ is strictly an ϵ-covering under $\|\cdot\|_2$. This topological inclusion directly yields $\mathcal{N}(\mathcal{M}, \epsilon, \|\cdot\|_2) \leq \mathcal{N}(\mathcal{M}, \epsilon, d_{\mathcal{M}})$. The following lemma establishes the bounds for these covering numbers and the local metric equivalence.

Lemma 2. *Let $\mathcal{M} \subseteq [0,1]^{\bar{d}}$ be a d-dimensional compact and connected Riemannian manifold with reach $\tau > 0$.*

1. *(Metric Equivalence [14, Lemma 3]) For any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}$ such that $\|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq \tau/2$, the intrinsic geodesic distance satisfies*

$$d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y}) \leq \tau \left(1 - \sqrt{1 - \frac{2\|\boldsymbol{x} - \boldsymbol{y}\|_2}{\tau}} \right) \leq 2\|\boldsymbol{x} - \boldsymbol{y}\|_2. \tag{5}$$

2. *(Covering Number Bound [31]) For any $0 < \epsilon \leq \tau/2$, the covering numbers of $\mathcal{M}$ are bounded by*

$$\mathcal{N}(\mathcal{M}, \epsilon, \|\cdot\|_2) \leq \mathcal{N}(\mathcal{M}, \epsilon, d_{\mathcal{M}}) \leq C_{\mathcal{M}} \epsilon^{-d}, \tag{6}$$

where $C_{\mathcal{M}} = 3^d Vol(\mathcal{M}) d^{d/2}$, and $Vol(\mathcal{M})$ is the volume of $\mathcal{M}$.

With the local metric equivalence and intrinsic covering number bound of manifolds established, we proceed to the Softmax POU approximation scheme on Riemannian manifolds. The proof controls the near-center error using the intrinsic Hölder continuity, and bounds the far-field error through the exponential decay of the ambient Softmax weights.

Lemma 3 (Softmax POU Approximation on Manifolds)**.** *Let $\mathcal{M} \subseteq [0,1]^{\bar{d}}$ be a compact and connected d-dimensional Riemannian manifold with reach $\tau > 0$, and let $g : \mathcal{M} \to \mathbb{R}$ be an α-Hölder continuous function with constant C_H. For any target accuracy ε satisfying $0 < \varepsilon \leq \min\left\{ \frac{1}{e}, 8C_H \left(\frac{\tau}{4}\right)^{\alpha} \right\}$, let the covering radius be $r_g = (\varepsilon/(8C_H))^{1/\alpha}$. There exists a finite set of C_g centers $\{\boldsymbol{c}_i\}_{i=1}^{C_g} \subset \mathcal{M}$ forming an open r_g-covering of the manifold (i.e., $\mathcal{M} \subseteq \bigcup_{i=1}^{C_g} \mathcal{B}(\boldsymbol{c}_i, r_g)$) with $C_g \leq C_{\mathcal{M}} (8C_H)^{d/\alpha} \varepsilon^{-d/\alpha}$. By setting the scaling parameter*

$$M_g = C_M \varepsilon^{-2/\alpha} \log \frac{1}{\varepsilon}, \quad \text{where } C_M = \frac{(8C_H)^{2/\alpha}}{3} \left(\log\left(4\|g\|_{\infty} C_{\mathcal{M}} (8C_H)^{d/\alpha}\right) + \frac{d+\alpha}{\alpha} \right),$$

the ambient Softmax POU approximation

$$\widehat{g}(\boldsymbol{x}) = \sum_{i=1}^{C_g} \beta_i(\boldsymbol{x}) g(\boldsymbol{c}_i),$$

with

$$\beta_i(\boldsymbol{x}) = \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g} \exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2)\right)} = \frac{\exp\left(2M_g \langle \boldsymbol{x}, \boldsymbol{c}_i \rangle - M_g \|\boldsymbol{c}_i\|_2^2\right)}{\sum_{k=1}^{C_g} \exp\left(2M_g \langle \boldsymbol{x}, \boldsymbol{c}_k \rangle - M_g \|\boldsymbol{c}_k\|_2^2\right)}, \tag{7}$$

achieves the uniform approximation bound

$$\sup_{\boldsymbol{x} \in \mathcal{M}} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \varepsilon.$$

Proof of Lemma 3. To construct the approximation, we first establish a finite covering of the data manifold $\mathcal{M}$ by open Euclidean balls $\{\mathcal{B}(\boldsymbol{c}_i, r_g)\}_{i=1}^{C_g}$ of radius $r_g > 0$, with centers $\boldsymbol{c}_i \in \mathcal{M}$. By the covering number bound in Lemma 2, such an open covering exists with the number of centers bounded by $C_g \leq C_{\mathcal{M}} r_g^{-d}$. For any $\boldsymbol{x} \in \mathcal{M}$, let $i^*(\boldsymbol{x}) = \operatorname{argmin}_{i\in[C_g]} \|\boldsymbol{x} - \boldsymbol{c}_i\|_2$. Since these open balls cover $\mathcal{M}$, we are guaranteed that $\|\boldsymbol{x} - \boldsymbol{c}_{i^*}\|_2 < r_g$. Notice that $\beta_i(\boldsymbol{x}) = \frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x}-\boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2-\|\boldsymbol{x}-\boldsymbol{c}_k\|_2^2)\right)}$, using the property that $\sum_{i=1}^{C_g}\beta_i(\boldsymbol{x}) = 1$, by decomposing the approximation error into near and far components at a Euclidean radius of $2r_g$, we have

$$
\begin{aligned}
|\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| &= \left|\sum_{i=1}^{C_g}\beta_i(\boldsymbol{x})\big(g(\boldsymbol{c}_i) - g(\boldsymbol{x})\big)\right| \\
&\leq \underbrace{\sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2\leq 2r_g}\beta_i(\boldsymbol{x})|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)|}_{(I)} + \underbrace{\sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2> 2r_g}\beta_i(\boldsymbol{x})|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)|}_{(II)}.
\end{aligned}
$$

For the near-center terms (I), we can bound the error using the intrinsic Hölder continuity. For sufficiently small ε such that $2r_g \leq \tau/2$, by Lemma 2, we have $d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{c}_i) \leq 2\|\boldsymbol{x} - \boldsymbol{c}_i\|_2 \leq 4r_g$. Consequently, the intrinsic α-Hölder continuity yields

$$
|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \leq C_H(d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{c}_i))^\alpha \leq C_H(4r_g)^\alpha \leq 4C_H r_g^\alpha,
$$

since $4^\alpha \leq 4$ for $\alpha \in (0, 1]$. Thus

$$
(I) \leq 4C_H r_g^\alpha \sum_i \beta_i(\boldsymbol{x}) \leq 4C_H r_g^\alpha.
$$

For the far-center tail terms (II), the condition $\|\boldsymbol{x} - \boldsymbol{c}_i\|_2 > 2r_g$ implies $r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2 < -3r_g^2$. Since $\|\boldsymbol{x} - \boldsymbol{c}_{i^*}\|_2 \leq r_g$, the denominator of the ambient Softmax POU $\beta_i(\boldsymbol{x})$ is bounded below

$$
\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2)\right) \geq \exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_{i^*}\|_2^2)\right) \geq \exp(0) = 1.
$$

Applying the global bound $|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \leq 2\|g\|_\infty$, the term (II) is bounded by

$$
\begin{aligned}
(II) &= \sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2>2r_g}\frac{\exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2)\right)}{\sum_{k=1}^{C_g}\exp\left(M_g(r_g^2 - \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2)\right)}|g(\boldsymbol{x}) - g(\boldsymbol{c}_i)| \\
&\leq \sum_{i:\|\boldsymbol{x}-\boldsymbol{c}_i\|_2>2r_g}\exp(-3M_g r_g^2)\cdot 2\|g\|_\infty \\
&\leq 2C_g\|g\|_\infty\exp(-3M_g r_g^2).
\end{aligned}
$$

Combining (I) and (II) yields

$$
\sup_{\boldsymbol{x}\in\mathcal{M}}|\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \leq 4C_H r_g^\alpha + 2C_g\|g\|_\infty\exp(-3M_g r_g^2). \tag{8}
$$

Setting $r_g = (\varepsilon/(8C_H))^{1/\alpha}$ yields $(I) \leq \varepsilon/2$. By Lemma 2, the number of Euclidean balls of radius r_g required to cover $\mathcal{M}$ is bounded by $C_g \leq C_{\mathcal{M}} r_g^{-d}$, where $C_{\mathcal{M}} = 3^d\text{Vol}(\mathcal{M})d^{d/2}$. Substituting our choice of r_g, we obtain

$$
C_g \leq C_{\mathcal{M}}\left(\left(\frac{\varepsilon}{8C_H}\right)^{1/\alpha}\right)^{-d} = C_{\mathcal{M}}(8C_H)^{d/\alpha}\varepsilon^{-\frac{d}{\alpha}}.
$$

Finally, to guarantee $(II) \leq \varepsilon/2$, the scaling parameter must satisfy $M_g \geq (3r_g^2)^{-1}\log(4\|g\|_\infty C_g/\varepsilon)$. Substituting the explicit formulas for r_g and the upper bound for C_g, we

require

$$M_g \geq \frac{(8C_H)^{2/\alpha}}{3}\varepsilon^{-\frac{2}{\alpha}}\log\left(4\|g\|_\infty C_{\mathcal{M}}(8C_H)^{d/\alpha}\varepsilon^{-\frac{d+\alpha}{\alpha}}\right)$$
$$= \frac{(8C_H)^{2/\alpha}}{3}\varepsilon^{-\frac{2}{\alpha}}\left(\log\left(4\|g\|_\infty C_{\mathcal{M}}(8C_H)^{d/\alpha}\right) + \frac{d+\alpha}{\alpha}\log\frac{1}{\varepsilon}\right).$$

Assuming $\varepsilon \leq 1/e$ so that $\log(1/\varepsilon) \geq 1$, we can upper bound the bracketed sum by factoring out $\log(1/\varepsilon)$. Defining M_g as

$$M_g = \frac{(8C_H)^{2/\alpha}}{3}\left(\log\left(4\|g\|_\infty C_{\mathcal{M}}(8C_H)^{d/\alpha}\right) + \frac{d+\alpha}{\alpha}\right)\varepsilon^{-\frac{2}{\alpha}}\log\frac{1}{\varepsilon}$$

satisfies this condition, ensuring $(II) \leq \varepsilon/2$. This achieves the target uniform error $\sup_{\boldsymbol{x}\in\mathcal{M}}|\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \varepsilon$ and completes the proof. □

A.3 Constructing Transformers for Softmax POU approximation: fundamental lemmas

This subsection details the the structural lemmas (Lemmas 4, 5, 6, and 7) and their proofs. These proofs demonstrate how the native softmax attention mechanism in Transformers can realize our Softmax POU approximation framework. The pre-processing step embeds $\boldsymbol{x}$ and an indicator exclusively into the first token, zero-pads the remaining sequence, and appends sinusoidal positional encodings across all P tokens.

Lemma 4 (Pre-processing Step)**.** *Let $\boldsymbol{x} \in [0,1]^d$ be the input vector, $P \geq 1$ be the sequence length, and $D = d+4$ be the embedding dimension. There exists a pre-processing operator $\mathcal{P} : \mathbb{R}^d \to \mathbb{R}^{D\times P}$ with the absolute values of the parameters bounded by 1, such that its output $\boldsymbol{Z}_0 = \mathcal{P}(\boldsymbol{x})$ is*

$$\boldsymbol{Z}_0 = \begin{bmatrix} \boldsymbol{x} & \mathbf{0} & \cdots & \mathbf{0} \\ 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_P) \\ \cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_P) \end{bmatrix} \in \mathbb{R}^{D\times P},$$

where $\theta_j = \frac{2\pi j}{P}$ for $j \in [P]$.

Proof of Lemma 4. Let $\boldsymbol{e}_1 = [1,0,\ldots,0]^\top \in \mathbb{R}^P$ be the indicator vector for the first token. We construct the pre-processing operator as

$$\mathcal{P}(\boldsymbol{x}) = (\boldsymbol{W}_E\boldsymbol{x} + \boldsymbol{b}_E)\boldsymbol{e}_1^\top + \boldsymbol{P}.$$

We define the embedding weight matrix $\boldsymbol{W}_E \in \mathbb{R}^{D\times d}$ and the bias vector $\boldsymbol{b}_E \in \mathbb{R}^D$ as

$$\boldsymbol{W}_E = \begin{bmatrix} \boldsymbol{I}_d \\ \mathbf{0}_{1\times d} \\ \mathbf{0}_{1\times d} \\ \mathbf{0}_{2\times d} \end{bmatrix}, \quad \boldsymbol{b}_E = \begin{bmatrix} \mathbf{0}_{d\times 1} \\ 1 \\ 0 \\ \mathbf{0}_{2\times 1} \end{bmatrix},$$

and the positional encoding matrix $\boldsymbol{P} \in \mathbb{R}^{D\times P}$ as

$$\boldsymbol{P} = \begin{bmatrix} & \mathbf{0}_{d\times P} & \\ & \mathbf{0}_{1\times P} & \\ & \mathbf{0}_{1\times P} & \\ \sin(\theta_1) & \ldots & \sin(\theta_P) \\ \cos(\theta_1) & \ldots & \cos(\theta_P) \end{bmatrix},$$

where $\theta_j = \frac{2\pi j}{P}$ for $j \in [P]$. Therefore, we have

$$\boldsymbol{Z}_0 = \mathcal{P}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x} & \mathbf{0} & \cdots & \mathbf{0} \\ 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_P) \\ \cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_P) \end{bmatrix} \in \mathbb{R}^{D\times P},$$

thus we complete the proof. □

The first MHA layer extracts two affine transformatioin features, while the exact angular identity θ_j of the sinusoidal positional encodings remains invariant up to a uniform scalar scaling.

Lemma 5 (Parallel Affine Transformations via MHA)**.** *Let $\boldsymbol{Z}_0 \in \mathbb{R}^{D\times P}$ ($D = d + 4$) as defined in Lemma 4. For target affine functions $T_h(\boldsymbol{x}) = \boldsymbol{\xi}_h^\top \boldsymbol{x} + b_h$ and $R_h(\boldsymbol{x}) = \boldsymbol{\zeta}_h^\top \boldsymbol{x} + c_h$ ($h \in [P]$), let $B_T := \max_h \|T_h\|_\infty$, $B_R := \max_h \|R_h\|_\infty$, and $B_{\mathcal{A}_1} := \max_h \max\{\|\boldsymbol{\xi}_h\|_\infty, \|\boldsymbol{\zeta}_h\|_\infty, |b_h|, |c_h|\}$. There exists an MHA layer $\mathcal{A}_1 : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with $H = P + 2$ heads, $d_v = d_k = 2$, and scaling factor $M > 0$, yielding output $\widehat{\boldsymbol{Z}}_1 = \mathcal{A}_1(\boldsymbol{Z}_0) \in \mathbb{R}^{D\times P}$ whose j-th column is*

$$(\widehat{\boldsymbol{Z}}_1)_{:,j} = \begin{bmatrix} \widetilde{T}_j(\boldsymbol{x}) \\ \widetilde{R}_j(\boldsymbol{x}) \\ \mathbf{0}_{d-1} \\ \widetilde{I}_j \\ \lambda(M)\sin(\theta_j) \\ \lambda(M)\cos(\theta_j) \end{bmatrix}, \quad \forall j \in [P].$$

Let $c = 1 - \cos(2\pi/P) > 0$. The output components and parameter magnitude satisfy

1. ***Affine Outputs:*** *we have the approximation error*

$$|T_j(\boldsymbol{x}) - \widetilde{T}_j(\boldsymbol{x})| \le (P-1)B_T e^{-cM},$$
$$|R_j(\boldsymbol{x}) - \widetilde{R}_j(\boldsymbol{x})| \le (P-1)B_R e^{-cM}.$$

2. ***Indicator Output:*** $\widetilde{I}_1 = \eta(M) := \frac{e^M}{\sum_{l=1}^P e^{M\cos(\theta_l)}}$, *and*

$$0 \le 1 - \widetilde{I}_1 = 1 - \eta(M) \le (P-1)e^{-cM}, \qquad (j = 1)$$
$$0 \le \widetilde{I}_j \le e^{-cM}. \qquad (j \ne 1)$$

3. ***PE Output:*** *the scaling coefficient $\lambda(M)$ satisfies*

$$\lambda(M) := \frac{\sum_{k=1}^P e^{M\cos(\theta_k)}\cos(\theta_k)}{\sum_{l=1}^P e^{M\cos(\theta_l)}}, \quad 0 \le 1 - \lambda(M) \le 2(P-1)e^{-cM}.$$

4. ***Parameter Bounds:*** *the maximum parameter magnitude $M_{\mathcal{A}_1}$ satisfy*

$$M_{\mathcal{A}_1} \le \max\left\{1, M, (1 + (P-1)e^{-cM})B_{\mathcal{A}_1}\right\}.$$

Proof of Lemma 5. We set the uniform dimension for query, key, and value matrices to $d_k = 2$ and $d_v = 2$.

Step 1: Affine Heads ($h \in [P]$). We construct the first P heads in the first encoder block to selectively route the affine transformations. We define the 2D rotation matrix $\boldsymbol{R}(\theta)$ as

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Before defining the attention matrices, we introduce a deterministic scalar $\eta(M)$ representing the exact peak attention weight, that is

$$\eta(M) := \frac{e^M}{\sum_{l=1}^P e^{M\cos(\theta_l)}}.$$

To eliminate the influence of the attention matrix while realizing affine transformations, we preemptively rescale the value matrix by $\eta(M)^{-1}$. Adhering to the uniform value dimension $d_v = 2$, we stack the weights and biases of both affine transformations. The query, key, and value matrices

($\boldsymbol{Q}_1^h, \boldsymbol{K}_1^h, \boldsymbol{V}_1^h \in \mathbb{R}^{2\times D}$) are defined as

$$\boldsymbol{Q}_1^h = [\mathbf{0}_{2\times d} \quad \mathbf{0}_{2\times 2} \quad \boldsymbol{R}(\theta_1 - \theta_h)] \implies (\boldsymbol{Q}_1^h \boldsymbol{Z}_0)_{:,j} = \begin{bmatrix}\sin(\theta_j + \theta_1 - \theta_h)\\ \cos(\theta_j + \theta_1 - \theta_h)\end{bmatrix},$$

$$\boldsymbol{K}_1^h = [\mathbf{0}_{2\times d} \quad \mathbf{0}_{2\times 2} \quad M\boldsymbol{I}_2] \implies (\boldsymbol{K}_1^h \boldsymbol{Z}_0)_{:,k} = \begin{bmatrix}M\sin(\theta_k)\\ M\cos(\theta_k)\end{bmatrix},$$

$$\boldsymbol{V}_1^h = \frac{1}{\eta(M)}\begin{bmatrix}\boldsymbol{\xi}_h^\top & b_h & \mathbf{0}_{1\times 3}\\ \boldsymbol{\zeta}_h^\top & c_h & \mathbf{0}_{1\times 3}\end{bmatrix}$$
$$\implies (\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,k} = \frac{1}{\eta(M)}\begin{bmatrix}\boldsymbol{\xi}_h^\top \boldsymbol{x} + b_h\\ \boldsymbol{\zeta}_h^\top \boldsymbol{x} + c_h\end{bmatrix}\delta_{k,1} = \frac{1}{\eta(M)}\begin{bmatrix}T_h(\boldsymbol{x})\\ R_h(\boldsymbol{x})\end{bmatrix}\delta_{k,1}.$$

Following the attention mechanism, the unnormalized score $s_{k,j}^h$ is defined as the inner product between the k-th key column and the j-th query column, that is,

$$s_{k,j}^h = (\boldsymbol{K}_1^h \boldsymbol{Z}_0)_{:,k}^\top (\boldsymbol{Q}_1^h \boldsymbol{Z}_0)_{:,j} = M\cos\big(\theta_k - (\theta_j + \theta_1 - \theta_h)\big).$$

Consequently, the (k,j)-th entry of the attention matrix $\boldsymbol{A}_1^h \in \mathbb{R}^{P\times P}$ is computed via the column-wise softmax function, we have

$$(\boldsymbol{A}_1^h)_{k,j} = \frac{\exp(s_{k,j}^h)}{S_j}, \quad \text{where } S_j = \sum_{l=1}^P \exp(s_{l,j}^h).$$

Let $c = 1 - \cos(2\pi/P) > 0$ denote the spectral gap. For any $\theta_a \neq \theta_b$, we have $\cos(\theta_a - \theta_b) \leq 1 - c$. We analyze the columns of $\boldsymbol{A}_1^h$ (indexed by the query token j) by bounding the numerator and the denominator S_j in the following.

- **Target Column ($j = h$):** The relative phase shift perfectly aligns the attention peak with the source key $k = 1$. Substituting $j = h$ yields $s_{k,h}^h = M\cos(\theta_k - \theta_1)$. For the source key ($k = 1$), the score reaches its exact maximum $s_{1,h}^h = M\cos(0) = M$. For the denominator $S_h = \sum_{l=1}^P e^{M\cos(\theta_l - \theta_1)}$, subtracting the constant phase θ_1 merely applies a cyclic permutation to the uniform grid. Due to the shift-invariance of the periodic grid, we have $S_h = \sum_{l=1}^P e^{M\cos(\theta_l)}$. Thus, the attention weight for the source key is

$$(\boldsymbol{A}_1^h)_{1,h} = \frac{e^{s_{1,h}^h}}{S_h} = \frac{e^M}{\sum_{l=1}^P e^{M\cos(\theta_l)}} = \eta(M),$$

  therefore, we have

$$(\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,1}(\boldsymbol{A}_1^h)_{1,h} = \eta(M)\cdot\frac{1}{\eta(M)}\begin{bmatrix}T_h(\boldsymbol{x})\\ R_h(\boldsymbol{x})\end{bmatrix} = \begin{bmatrix}T_h(\boldsymbol{x})\\ R_h(\boldsymbol{x})\end{bmatrix}.$$

  Because $(\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,k} = \mathbf{0}$ for $k \neq 1$, we get

$$(\boldsymbol{V}_1^h \boldsymbol{Z}_0)(\boldsymbol{A}_1^h)_{:,h} = \sum_{k=1}^P (\boldsymbol{A}_1^h)_{k,h}(\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,k} = (\boldsymbol{A}_1^h)_{1,h}(\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,1} = \begin{bmatrix}T_h(\boldsymbol{x})\\ R_h(\boldsymbol{x})\end{bmatrix}.$$

- **Other Columns ($j \neq h$):** The query perfectly matches a shifted key position k^* satisfying $\theta_{k^*} \equiv \theta_j + \theta_1 - \theta_h$. Since the target differs ($j \neq h$), the attention peak shifts away from the source token ($k^* \neq 1$). The unnormalized attention score at the source key ($k = 1$) is

$$s_{1,j}^h = M\cos\big(\theta_1 - (\theta_j + \theta_1 - \theta_h)\big) = M\cos(\theta_h - \theta_j) \leq M(1-c).$$

  We denote the output for these non-target columns as $(\boldsymbol{V}_1^h \boldsymbol{Z}_0)(\boldsymbol{A}_1^h)_{:,j} := [\epsilon_{h,j}, \widetilde{\epsilon}_{h,j}]^\top$, then

$$\begin{bmatrix}\epsilon_{h,j}\\ \widetilde{\epsilon}_{h,j}\end{bmatrix} = (\boldsymbol{V}_1^h \boldsymbol{Z}_0)(\boldsymbol{A}_1^h)_{:,j} = \sum_{k=1}^P (\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,k}(\boldsymbol{A}_1^h)_{k,j}$$
$$= (\boldsymbol{V}_1^h \boldsymbol{Z}_0)_{:,1}(\boldsymbol{A}_1^h)_{1,j} = \frac{(\boldsymbol{A}_1^h)_{1,j}}{\eta(M)}\begin{bmatrix}T_h(\boldsymbol{x})\\ R_h(\boldsymbol{x})\end{bmatrix}.$$

Due to the exact shift-invariance, $S_j = \sum_{l=1}^{P} e^{M\cos(\theta_l-\theta_j)} = \sum_{l=1}^{P} e^{M\cos(\theta_l)} \equiv S_h$, thus

$$\epsilon_{h,j} = \frac{e^{s^h_{1,j}}/S_j}{e^M/S_h} T_h(\boldsymbol{x}) = e^{s^h_{1,j}-M} T_h(\boldsymbol{x}) \implies |\epsilon_{h,j}| \le e^{M(1-c)-M} B_T = e^{-cM} B_T,$$

$$\widetilde{\epsilon}_{h,j} = \frac{e^{s^h_{1,j}}/S_j}{e^M/S_h} R_h(\boldsymbol{x}) = e^{s^h_{1,j}-M} R_h(\boldsymbol{x}) \implies |\widetilde{\epsilon}_{h,j}| \le e^{M(1-c)-M} B_R = e^{-cM} B_R.$$

Step 2: Indicator & Positional Encoding Heads ($h \in \{P+1, P+2\}$). We dedicate Head $P+1$ to preserve the indicator $\delta_{j,1}$ and Head $P+2$ to preserve the positional encodings. The query, key, and value matrices ($\boldsymbol{Q}_1^h, \boldsymbol{K}_1^h, \boldsymbol{V}_1^h \in \mathbb{R}^{2\times D}$) are defined as

$$\boldsymbol{Q}_1^h = [\boldsymbol{0}_{2\times d} \quad \boldsymbol{0}_{2\times 2} \quad \boldsymbol{I}_2] \implies (\boldsymbol{Q}_1^h \boldsymbol{Z}_0)_{:,j} = \begin{bmatrix}\sin(\theta_j)\\ \cos(\theta_j)\end{bmatrix}, \quad \forall h \in \{P+1, P+2\},$$

$$\boldsymbol{K}_1^h = [\boldsymbol{0}_{2\times d} \quad \boldsymbol{0}_{2\times 2} \quad M\boldsymbol{I}_2] \implies (\boldsymbol{K}_1^h \boldsymbol{Z}_0)_{:,k} = \begin{bmatrix}M\sin(\theta_k)\\ M\cos(\theta_k)\end{bmatrix}, \quad \forall h \in \{P+1, P+2\},$$

$$\boldsymbol{V}_1^{P+1} = \begin{bmatrix}\boldsymbol{0}_{1\times d} & 1 & \boldsymbol{0}_{1\times 3}\\ \boldsymbol{0}_{1\times d} & 0 & \boldsymbol{0}_{1\times 3}\end{bmatrix} \implies (\boldsymbol{V}_1^{P+1} \boldsymbol{Z}_0)_{:,k} = \begin{bmatrix}\delta_{k,1}\\ 0\end{bmatrix},$$

$$\boldsymbol{V}_1^{P+2} = [\boldsymbol{0}_{2\times d} \quad \boldsymbol{0}_{2\times 2} \quad \boldsymbol{I}_2] \implies (\boldsymbol{V}_1^{P+2} \boldsymbol{Z}_0)_{:,k} = \begin{bmatrix}\sin(\theta_k)\\ \cos(\theta_k)\end{bmatrix}.$$

The shared unnormalized attention score is given by

$$s^{\text{id}}_{k,j} = (\boldsymbol{K}_1^h \boldsymbol{Z}_0)^\top_{:,k} (\boldsymbol{Q}_1^h \boldsymbol{Z}_0)_{:,j} = M\big(\sin(\theta_j)\sin(\theta_k) + \cos(\theta_j)\cos(\theta_k)\big) = M\cos(\theta_j - \theta_k).$$

The corresponding attention weight is $(\boldsymbol{A}^{\text{id}})_{k,j} = \frac{\exp(M\cos(\theta_j-\theta_k))}{S^{\text{id}}}$. Due to the symmetry of the uniform periodic grid, the normalization denominator S^{id} is invariant to the query index j. By applying a periodic shift to the summation index, it simplifies to

$$S^{\text{id}} = \sum_{l=1}^{P} \exp\big(M\cos(\theta_j - \theta_l)\big) = \sum_{l=1}^{P} \exp\big(M\cos(\theta_l)\big).$$

For the indicator head ($h = P+1$), to align with $d_v = 2$, we pad the indicator value with a zero row, making the value vector $[\delta_{k,1}, 0]^\top$. Thus, the output for the j-th token is $[\widetilde{I}_j, 0]^\top$, where the first component $\widetilde{I}_j$ evaluates to

$$\widetilde{I}_j = \sum_{k=1}^{P} (\boldsymbol{A}^{\text{id}})_{k,j}\delta_{k,1} = (\boldsymbol{A}^{\text{id}})_{1,j} = \frac{e^{M\cos(\theta_j-\theta_1)}}{\sum_{l=1}^{P} e^{M\cos(\theta_l)}}.$$

For the source token ($j = 1$), the numerator is e^M. We define this exact output as a deterministic scalar coefficient $\eta(M)$ that

$$\widetilde{I}_1 = \eta(M) := \frac{e^M}{\sum_{l=1}^{P} e^{M\cos(\theta_l)}}.$$

By bounding the $P-1$ off-diagonal terms in the denominator by $e^{M(1-c)}$, we get

$$\eta(M) \ge \frac{e^M}{e^M + (P-1)e^{M(1-c)}} \implies 0 \le 1 - \eta(M) \le (P-1)e^{-cM}.$$

For non-source tokens ($j \ne 1$), the numerator is bounded by $e^{M(1-c)}$ while the denominator is at least e^M, we have

$$\widetilde{I}_j \le \frac{e^{M(1-c)}}{e^M} = e^{-cM}.$$

This deterministic scalar $\eta(M)$ allows the subsequent point-wise FFN layer to restore the indicator to 1.

For the PE head ($h = P + 2$), the value vector is $[\sin(\theta_k), \cos(\theta_k)]^\top$. By substituting the relative phase $\phi_k = \theta_k - \theta_j$, the output for the j-th token becomes

$$\sum_{k=1}^{P} (\boldsymbol{A}^{\text{id}})_{k,j} \begin{bmatrix} \sin(\theta_k) \\ \cos(\theta_k) \end{bmatrix} = \frac{1}{S^{\text{id}}} \sum_{k=1}^{P} e^{M\cos(\phi_k)} \begin{bmatrix} \sin(\phi_k + \theta_j) \\ \cos(\phi_k + \theta_j) \end{bmatrix}$$
$$= \frac{1}{S^{\text{id}}} \sum_{k=1}^{P} e^{M\cos(\phi_k)} \begin{bmatrix} \sin(\phi_k)\cos(\theta_j) + \cos(\phi_k)\sin(\theta_j) \\ \cos(\phi_k)\cos(\theta_j) - \sin(\phi_k)\sin(\theta_j) \end{bmatrix}.$$

Since subtracting the grid point θ_j applies a cyclic permutation, the relative phases reconstruct the identical equispaced grid of base angles. Specifically, evaluated as an unordered set modulo 2π, we have the set equality: $\Phi = \{\phi_k \pmod{2\pi}\}_{k=1}^{P} = \left\{\frac{2\pi(k-1)}{P}\right\}_{k=1}^{P} = \{\theta_k\}_{k=1}^{P}$. Defining the summand $g(\phi) = e^{M\cos(\phi)}\sin(\phi)$, we partition the grid Φ into self-conjugate poles $\Phi_0 = \Phi \cap \{\pi, 2\pi\}$ (where $\pi \in \Phi$ if and only if P is even) and the remaining conjugate pairs $\{\phi, 2\pi - \phi\}$. For $\phi \in \Phi_0$, the term disappears trivially since $g(0) = g(\pi) = 0$. For any conjugate pair, we have

$$g(2\pi - \phi) + g(\phi) = e^{M\cos(\phi)}\big(-\sin(\phi) + \sin(\phi)\big) = 0.$$

Thus, regardless of the parity of P, the global summation becomes

$$\sum_{k=1}^{P} e^{M\cos(\phi_k)}\sin(\phi_k) = \sum_{\phi\in\Phi_0} g(\phi) + \frac{1}{2}\sum_{\phi\notin\Phi_0} \big(g(\phi) + g(2\pi - \phi)\big) = 0.$$

Hence, utilizing the established set equality $\Phi = \{\theta_l\}_{l=1}^{P}$, the output for j-th token simplifies to a scaled version of the input positional encoding, that is

$$\left(\frac{\sum_{l=1}^{P} e^{M\cos(\theta_l)}\cos(\theta_l)}{\sum_{l=1}^{P} e^{M\cos(\theta_l)}}\right) \begin{bmatrix} \sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix} = \lambda(M) \begin{bmatrix} \sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix}.$$

Considering $1 - \lambda(M) = \frac{\sum_{l=1}^{P} e^{M\cos(\theta_l)}(1-\cos(\theta_l))}{\sum_{l=1}^{P} e^{M\cos(\theta_l)}}$, for the peak term where $\theta_l = 0$ (corresponding to $l = P$), the numerator term is $e^M(1 - \cos(0)) = 0$. For the remaining $P - 1$ terms ($l \neq P$), we have $e^{M\cos(\theta_l)} \leq e^{M(1-c)}$ and $(1 - \cos(\theta_l)) \leq 2$. Lower bounding the denominator by the single peak term e^M, we obtain

$$0 \leq 1 - \lambda(M) \leq \frac{(P-1)\cdot e^{M(1-c)}\cdot 2}{e^M} = 2(P-1)e^{-cM}.$$

Thus, positional encodings are preserved with a uniform scaling coefficient $\lambda(M)$, allowing easy restoration in the following point-wise FFN layer.

Step 3: Output Projection and Error Bounds. Concatenating the outputs of all $P + 2$ heads yields the intermediate matrix $\boldsymbol{C} \in \mathbb{R}^{2(P+2)\times P}$ in the following form

$$\boldsymbol{C} = \begin{bmatrix} T_1(\boldsymbol{x}) & \epsilon_{1,2} & \dots & \epsilon_{1,P} \\ R_1(\boldsymbol{x}) & \widetilde{\epsilon}_{1,2} & \dots & \widetilde{\epsilon}_{1,P} \\ \epsilon_{2,1} & T_2(\boldsymbol{x}) & \dots & \epsilon_{2,P} \\ \widetilde{\epsilon}_{2,1} & R_2(\boldsymbol{x}) & \dots & \widetilde{\epsilon}_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{P,1} & \epsilon_{P,2} & \dots & T_P(\boldsymbol{x}) \\ \widetilde{\epsilon}_{P,1} & \widetilde{\epsilon}_{P,2} & \dots & R_P(\boldsymbol{x}) \\ \widetilde{I}_1 & \widetilde{I}_2 & \dots & \widetilde{I}_P \\ 0 & 0 & \dots & 0 \\ \lambda(M)\sin(\theta_1) & \lambda(M)\sin(\theta_2) & \dots & \lambda(M)\sin(\theta_P) \\ \lambda(M)\cos(\theta_1) & \lambda(M)\cos(\theta_2) & \dots & \lambda(M)\cos(\theta_P) \end{bmatrix}.$$

By introducing the block identity matrix $[\boldsymbol{I}_2 \quad \dots \quad \boldsymbol{I}_2] \in \mathbb{R}^{2\times 2P}$, we choose the output projection matrix $\boldsymbol{W}^O \in \mathbb{R}^{D\times 2(P+2)}$ as

$$\boldsymbol{W}^O = \begin{bmatrix} [\boldsymbol{I}_2 \quad \dots \quad \boldsymbol{I}_2] & \boldsymbol{0}_{2\times 2} & \boldsymbol{0}_{2\times 2} \\ \boldsymbol{0}_{(d-1)\times 2P} & \boldsymbol{0}_{(d-1)\times 2} & \boldsymbol{0}_{(d-1)\times 2} \\ \boldsymbol{0}_{1\times 2P} & [1 \quad 0] & \boldsymbol{0}_{1\times 2} \\ \boldsymbol{0}_{2\times 2P} & \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_2 \end{bmatrix},$$

the MHA output for the j-th token is

$$(\widehat{\boldsymbol{Z}}_1)_{:,j} = \boldsymbol{W}^O \boldsymbol{C}_{:,j} = \begin{bmatrix} \widetilde{T}_j(\boldsymbol{x}) \\ \widetilde{R}_j(\boldsymbol{x}) \\ \mathbf{0}_{d-1} \\ \widetilde{I}_j \\ \lambda(M) \begin{bmatrix} \sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix} \end{bmatrix},$$

where

$$\widetilde{T}_j(\boldsymbol{x}) = T_j(\boldsymbol{x}) + \sum_{h \neq j} \epsilon_{h,j}, \quad \widetilde{R}_j(\boldsymbol{x}) = R_j(\boldsymbol{x}) + \sum_{h \neq j} \widetilde{\epsilon}_{h,j}.$$

Given the uniform bounds $|\epsilon_{h,j}| \leq e^{-cM} B_T$ and $|\widetilde{\epsilon}_{h,j}| \leq e^{-cM} B_R$, we have

$$|\widetilde{T}_j(\boldsymbol{x}) - T_j(\boldsymbol{x})| = \Big| \sum_{h \neq j} \epsilon_{h,j} \Big| \leq \sum_{h \neq j} |\epsilon_{h,j}| \leq (P-1) e^{-cM} B_T,$$
$$|\widetilde{R}_j(\boldsymbol{x}) - R_j(\boldsymbol{x})| = \Big| \sum_{h \neq j} \widetilde{\epsilon}_{h,j} \Big| \leq \sum_{h \neq j} |\widetilde{\epsilon}_{h,j}| \leq (P-1) e^{-cM} B_R.$$

Step 4: Parameter Bounds. To bound the parameter magnitude, we evaluate the maximum absolute entry $\|\cdot\|_{\max}$ for each parameter matrix over all heads $h \in [P+2]$. The peak attention weight satisfies $\eta(M)^{-1} \leq \frac{e^M + (P-1)e^{M(1-c)}}{e^M} = 1 + (P-1)e^{-cM}$. It follows that the maximum entries are explicitly bounded by

$$\begin{aligned}
\max_h \|\boldsymbol{Q}_1^h\|_{\max} &\leq 1, \\
\max_h \|\boldsymbol{K}_1^h\|_{\max} &= M, \\
\max_h \|\boldsymbol{V}_1^h\|_{\max} &\leq \max\left\{1, \eta(M)^{-1} B_{\mathcal{A}_1}\right\} \leq \max\left\{1, (1 + (P-1)e^{-cM}) B_{\mathcal{A}_1}\right\}, \\
\|\boldsymbol{W}^O\|_{\max} &= 1.
\end{aligned}$$

Therefore, the maximum absolute value of any parameter in the MHA layer $\mathcal{A}_1$, denoted as $M_{\mathcal{A}_1}$, is bounded by

$$\begin{aligned}
M_{\mathcal{A}_1} &= \max\Big\{ \max_h \|\boldsymbol{Q}_1^h\|_{\max}, \max_h \|\boldsymbol{K}_1^h\|_{\max}, \max_h \|\boldsymbol{V}_1^h\|_{\max}, \|\boldsymbol{W}^O\|_{\max} \Big\} \\
&\leq \max\left\{1, M, (1 + (P-1)e^{-cM}) B_{\mathcal{A}_1}\right\}.
\end{aligned}$$

Thus, we complete the proof. □

The first point-wise FFN layer acts as a precise non-linear denoiser, leveraging the hard-thresholding property of ReLU activations to completely eliminate small errors and recover the canonical form and original sinusoidal positional encodings.

Lemma 6 (FFN Exact Thresholding and Restoration)**.** *Let $\widehat{\boldsymbol{Z}}_1 \in \mathbb{R}^{D \times P}$ (where $D = d + 4$) be the output matrix from the MHA layer defined in Lemma 5. Assume M is sufficiently large such that $(P+1)e^{-cM} < 1$. There exists a point-wise FFN layer $\mathcal{F}_1 : \mathbb{R}^{D \times P} \to \mathbb{R}^{D \times P}$ with hidden dimension $d_{ff} = 2D = 2d + 8$, such that its output $\boldsymbol{Z}_1 = \mathcal{F}_1(\widehat{\boldsymbol{Z}}_1) \in \mathbb{R}^{D \times P}$ takes the column-wise form*

$$(\boldsymbol{Z}_1)_{:,j} = \begin{bmatrix} \widetilde{T}_j(\boldsymbol{x}) \\ \widetilde{R}_j(\boldsymbol{x}) \\ \mathbf{0}_{d-1} \\ \delta_{j,1} \\ \sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix}, \quad \forall j \in [P].$$

The maximum parameter magnitude $M_{\mathcal{F}_1}$ satisfy

$$M_{\mathcal{F}_1} \leq \frac{1}{1 - 2Pe^{-cM}}.$$

Proof of Lemma 6. Let $w_{\text{ind}} = \frac{1}{\eta(M)-2e^{-cM}}$. We define the j-th column of the input as $\widehat{\boldsymbol{z}}_j = (\widehat{\boldsymbol{Z}}_1)_{:,j}$ and the j-th column of the output as $\boldsymbol{z}_j = (\boldsymbol{Z}_1)_{:,j}$. We construct the FFN applied to each column as $\boldsymbol{z}_j = \boldsymbol{W}_2\sigma(\boldsymbol{W}_1\widehat{\boldsymbol{z}}_j + \boldsymbol{b}_1)$.

To uniformly map the D-dimensional input across the $2D$ hidden neurons, we introduce a diagonal scaling matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{D\times D}$ defined as

$$\boldsymbol{\Lambda} = \operatorname{diag}\Big(\underbrace{1,\dots,1}_{d+1}, w_{\text{ind}}, \frac{1}{\lambda(M)}, \frac{1}{\lambda(M)}\Big).$$

We set the weights $\boldsymbol{W}_1 \in \mathbb{R}^{2D\times D}$, $\boldsymbol{W}_2 \in \mathbb{R}^{D\times 2D}$ and biases $\boldsymbol{b}_1 \in \mathbb{R}^{2D}$, $\boldsymbol{b}_2 \in \mathbb{R}^D$ as

$$\boldsymbol{W}_1 = \begin{bmatrix} \boldsymbol{I}_D \\ -\boldsymbol{I}_D \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} -2e^{-cM}\boldsymbol{e}_{d+2} \\ \boldsymbol{0}_D \end{bmatrix},$$
$$\boldsymbol{W}_2 = [\boldsymbol{\Lambda} \quad -\boldsymbol{\Lambda}], \quad \boldsymbol{b}_2 = \boldsymbol{0}_D,$$

where $\boldsymbol{e}_{d+2} \in \mathbb{R}^D$ is the standard basis vector for the $(d+2)$-th component.

Using the ReLU identity $x = \sigma(x) - \sigma(-x)$, we analyze the output column $\boldsymbol{z}_j$. By evaluating $\boldsymbol{z}_j = \boldsymbol{\Lambda}\sigma(\widehat{\boldsymbol{z}}_j - 2e^{-cM}\boldsymbol{e}_{d+2}) - \boldsymbol{\Lambda}\sigma(-\widehat{\boldsymbol{z}}_j)$ component-wise, we get

$$(\boldsymbol{z}_j)_{1:d+1} = \sigma((\widehat{\boldsymbol{z}}_j)_{1:d+1}) - \sigma(-(\widehat{\boldsymbol{z}}_j)_{1:d+1}) = (\widehat{\boldsymbol{z}}_j)_{1:d+1},$$
$$(\boldsymbol{z}_j)_{d+3:d+4} = \frac{1}{\lambda(M)}\big(\sigma((\widehat{\boldsymbol{z}}_j)_{d+3:d+4}) - \sigma(-(\widehat{\boldsymbol{z}}_j)_{d+3:d+4})\big) = \frac{1}{\lambda(M)}(\widehat{\boldsymbol{z}}_j)_{d+3:d+4}.$$

For the indicator neuron (index $d+2$), since $\widetilde{I}_j \geq 0$, the negative channel trivially evaluates to $\sigma(-\widetilde{I}_j) = 0$. Thus, substituting $(\widehat{\boldsymbol{z}}_j)_{d+2} = \widetilde{I}_j$ yields

$$(\boldsymbol{z}_j)_{d+2} = w_{\text{ind}} \cdot \sigma(\widetilde{I}_j - 2e^{-cM}) - 0$$
$$= \begin{cases} w_{\text{ind}} \cdot \sigma(\eta(M) - 2e^{-cM}) = 1, & \text{if } j = 1, \\ w_{\text{ind}} \cdot \sigma(\widetilde{I}_j - 2e^{-cM}) = 0, & \text{if } j \neq 1 \text{ (since } \widetilde{I}_j \leq e^{-cM}), \end{cases}$$

yielding $(\boldsymbol{z}_j)_{d+2} = \delta_{j,1}$. Concatenating all evaluated components, the exact restored output column is precisely $\boldsymbol{z}_j = [\widetilde{T}_j(\boldsymbol{x}), \widetilde{R}_j(\boldsymbol{x}), \boldsymbol{0}_{d-1}^\top, \delta_{j,1}, \sin(\theta_j), \cos(\theta_j)]^\top$.

Using the bounds $\lambda(M) \geq 1 - 2(P-1)e^{-cM}$ and $\eta(M) \geq 1 - (P-1)e^{-cM}$ from Lemma 5, the parameter magnitude is bounded by the maximum entry in $\boldsymbol{\Lambda}$ and $\boldsymbol{b}_1$:

$$M_{\mathcal{F}_1} = \max\Big\{1, \frac{1}{\lambda(M)}, \frac{1}{\eta(M) - 2e^{-cM}}, 2e^{-cM}\Big\}$$
$$\leq \max\Big\{\frac{1}{1-2(P-1)e^{-cM}}, \frac{1}{1-(P+1)e^{-cM}}\Big\} \leq \frac{1}{1-2Pe^{-cM}},$$

where the constants 1 and $2e^{-cM}$ are absorbed since the upper bound is strictly greater than 1. □

The second MHA layer aggregates local affine transformation features to a global approximant via softmax attention mechanism, utilizing $\widetilde{T}_j(\boldsymbol{x})$ as local spatial features and $\widetilde{R}_j(\boldsymbol{x})$ as local approximation expert features.

Lemma 7 (Softmax POU via MHA)**.** *Let $\boldsymbol{Z}_1 \in \mathbb{R}^{D\times P}$ (where $D = d+4$) be the output matrix from the FFN layer defined in Lemma 6. There exists a single-head MHA layer $\mathcal{A}_2 : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with $H = 1$ and $d_k = d_v = 1$, yielding output $\widehat{\boldsymbol{Z}}_2 = \mathcal{A}_2(\boldsymbol{Z}_1) \in \mathbb{R}^{D\times P}$ such that its first column ($k = 1$) evaluates to*

$$(\widehat{\boldsymbol{Z}}_2)_{:,1} = \begin{bmatrix} \widehat{g}_T(\boldsymbol{x}) \\ \boldsymbol{0}_{D-1} \end{bmatrix} = \widehat{g}_T(\boldsymbol{x})\boldsymbol{e}_1,$$

where

$$\widehat{g}_T(\boldsymbol{x}) := \sum_{j=1}^P \widetilde{\beta}_j(\boldsymbol{x})\widetilde{R}_j(\boldsymbol{x}), \quad \widetilde{\beta}_j(\boldsymbol{x}) := \frac{\exp(\widetilde{T}_j(\boldsymbol{x}))}{\sum_{l=1}^P \exp(\widetilde{T}_l(\boldsymbol{x}))},$$

and all other columns ($k \neq 1$) evaluate to constant vectors $\widetilde{R}(\boldsymbol{x})\boldsymbol{e}_1$, where $\widetilde{R}(\boldsymbol{x}) = \frac{1}{P}\sum_{j=1}^P \widetilde{R}_j(\boldsymbol{x})$. The maximum parameter magnitude $M_{\mathcal{A}_2}$ satisfy

$$M_{\mathcal{A}_2} = 1.$$

Proof of Lemma 7. We use a single attention head ($H = 1$) with $d_k = d_v = 1$. Let $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_{d+2} \in \mathbb{R}^D$ be the standard basis vectors. We define the projection matrices $\boldsymbol{Q}_2^1, \boldsymbol{K}_2^1, \boldsymbol{V}_2^1 \in \mathbb{R}^{1\times D}$ as

$$\begin{aligned}
\boldsymbol{Q}_2^1 = \boldsymbol{e}_{d+2}^\top &\implies (\boldsymbol{Q}_2^1 \boldsymbol{Z}_1)_{:,j} = \delta_{j,1}, \\
\boldsymbol{K}_2^1 = \boldsymbol{e}_1^\top &\implies (\boldsymbol{K}_2^1 \boldsymbol{Z}_1)_{:,k} = \widetilde{T}_k(\boldsymbol{x}), \\
\boldsymbol{V}_2^1 = \boldsymbol{e}_2^\top &\implies (\boldsymbol{V}_2^1 \boldsymbol{Z}_1)_{:,k} = \widetilde{R}_k(\boldsymbol{x}).
\end{aligned}$$

The unnormalized attention score $s_{k,j} = (\boldsymbol{K}_2^1 \boldsymbol{Z}_1)_{:,k}^\top (\boldsymbol{Q}_2^1 \boldsymbol{Z}_1)_{:,j}$ evaluates to

$$s_{k,j} = \widetilde{T}_k(\boldsymbol{x}) \cdot \delta_{j,1} = \begin{cases} \widetilde{T}_k(\boldsymbol{x}), & j = 1, \\ 0, & j \neq 1. \end{cases}$$

Applying the column-wise softmax function across the key indices $k \in [P]$ yields the attention weights $(\boldsymbol{A}_2^1)_{k,j}$. Since the normalization denominator is $S_j = \sum_{l=1}^P \exp(s_{l,j})$, we have

$$(\boldsymbol{A}_2^1)_{k,j} = \frac{\exp(s_{k,j})}{\sum_{l=1}^P \exp(s_{l,j})} = \begin{cases} \frac{\exp(\widetilde{T}_k(\boldsymbol{x}))}{\sum_{l=1}^P \exp(\widetilde{T}_l(\boldsymbol{x}))} = \widetilde{\beta}_k(\boldsymbol{x}), & j = 1, \\ \frac{\exp(0)}{\sum_{l=1}^P \exp(0)} = \frac{1}{P}, & j \neq 1. \end{cases}$$

The aggregated scalar output for the j-th token is

$$\boldsymbol{H}_{:,j} = \sum_{k=1}^P (\boldsymbol{A}_2^1)_{k,j} (\boldsymbol{V}_2^1 \boldsymbol{Z}_1)_{:,k} = \begin{cases} \sum_{k=1}^P \widetilde{\beta}_k(\boldsymbol{x}) \widetilde{R}_k(\boldsymbol{x}) = \widehat{g}_T(\boldsymbol{x}), & j = 1, \\ \frac{1}{P} \sum_{k=1}^P \widetilde{R}_k(\boldsymbol{x}) := \widetilde{R}(\boldsymbol{x}), & j \neq 1. \end{cases}$$

We define the output projection matrix $\boldsymbol{W}^O \in \mathbb{R}^{D\times 1}$ as $\boldsymbol{W}^O = \boldsymbol{e}_1$. The output columns are

$$(\widehat{\boldsymbol{Z}}_2)_{:,j} = \boldsymbol{W}^O \boldsymbol{H}_{:,j} = \begin{cases} \widehat{g}_T(\boldsymbol{x}) \boldsymbol{e}_1, & j = 1, \\ \widetilde{R}(\boldsymbol{x}) \boldsymbol{e}_1, & j \neq 1. \end{cases}$$

Since all entries in the projection matrices are canonical basis elements, the maximum parameter magnitude is bounded by $M_{\mathcal{A}_2} = 1$. Thus we complete the proof. □

B Proof of approximation results in Theorems 1 and 3

This section contains the comprehensive proofs for our main approximation guarantees. By assembling the foundational Transformer construction lemmas established in Subsection A.3, we construct the full architectures required for the target functions. We need the following lemma from Corollary A.7 in [11].

Lemma 8. *For any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, we have*

$$\|\mathrm{Softmax}(\boldsymbol{\theta}) - \mathrm{Softmax}(\boldsymbol{\theta}')\|_1 \leq 2\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\infty.$$

B.1 Approximation proof in Theorem 1

We provide the detailed construction and error bound analysis for Theorem 1. This formally establishes the uniform ε-approximation error for Hölder continuous functions defined on Euclidean domains.

Proof of Theorem 1. Applying Lemma 1 with target accuracy $\varepsilon/2$ yields P centers $\{\boldsymbol{c}_j\}_{j=1}^P \subset [0,1]^d$ with

$$P \leq \left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d (\varepsilon/2)^{-\frac{d}{\alpha}} = C_P \varepsilon^{-\frac{d}{\alpha}}, \quad \text{where } C_P = \left(\sqrt{d}(8C_H)^{1/\alpha}\right)^d.$$

Define the target affine features $T_j(\boldsymbol{x}) := 2M_g \boldsymbol{c}_j^\top \boldsymbol{x} - M_g \|\boldsymbol{c}_j\|_2^2$ and $R_j(\boldsymbol{x}) := g(\boldsymbol{c}_j)$. The corresponding Softmax POU approximation is given by

$$\widehat{g}(\boldsymbol{x}) = \sum_{j=1}^P \beta_j(\boldsymbol{x}) g(\boldsymbol{c}_j) = \sum_{j=1}^P \frac{\exp(T_j(\boldsymbol{x}))}{\sum_{l=1}^P \exp(T_l(\boldsymbol{x}))} g(\boldsymbol{c}_j). \tag{9}$$

By Lemma 1, this approximation satisfies

$$\sup_{\boldsymbol{x}\in[0,1]^d} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \le \varepsilon/2,$$

provided that

$$M_g = C_M \varepsilon^{-\frac{2}{\alpha}} \log \frac{2}{\varepsilon}, \quad \text{with } C_M = \frac{(8C_H)^{2/\alpha}}{3}\left(\log\left(4B\left(\sqrt{d}(4C_H)^{1/\alpha}\right)^d\right) + \frac{d+\alpha}{\alpha}\right).$$

Furthermore, since $\boldsymbol{x}, \boldsymbol{c}_j \in [0,1]^d$, the magnitudes of these target features are bounded by

$$B_T := \max_{j\in[P]} \|T_j\|_\infty \le 2M_g \sup_{\boldsymbol{x}\in[0,1]^d} |\boldsymbol{c}_j^\top \boldsymbol{x}| + M_g \|\boldsymbol{c}_j\|_2^2 \le 3M_g d,$$
$$B_R := \max_{j\in[P]} \|R_j\|_\infty = \max_{j\in[P]} |g(\boldsymbol{c}_j)| \le \|g\|_\infty \le B.$$

Step 1: Transformer Construction. By Lemma 4, the input $\boldsymbol{x}$ is mapped to the initial sequence matrix $\boldsymbol{Z}_0 = \mathcal{P}(\boldsymbol{x}) \in \mathbb{R}^{D\times P}$ structured as

$$\boldsymbol{Z}_0 = \begin{bmatrix} \boldsymbol{x} & \mathbf{0} & \cdots & \mathbf{0} \\ 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_P) \\ \cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_P) \end{bmatrix}.$$

For Encoder Block 1, applying the MHA layer $\mathcal{A}_1$ from Lemma 5 yields $\widehat{\boldsymbol{Z}}_1 = \mathcal{A}_1(\boldsymbol{Z}_0) \in \mathbb{R}^{D\times P}$, where the j-th column evaluates to

$$(\widehat{\boldsymbol{Z}}_1)_{:,j} = \left[\widetilde{T}_j(\boldsymbol{x}), \widetilde{g}(\boldsymbol{c}_j), \mathbf{0}_{d-1}^\top, \widetilde{I}_j, \lambda(M)\sin(\theta_j), \lambda(M)\cos(\theta_j)\right]^\top, \quad \forall j \in [P],$$

where $\widetilde{I}_1 = \eta(M) := \frac{e^M}{\sum_{l=1}^P e^{M\cos(\theta_l)}}$, and

$$0 \le 1 - \widetilde{I}_1 = 1 - \eta(M) \le (P-1)e^{-cM}, \qquad \text{for } j = 1,$$
$$0 \le \widetilde{I}_j \le e^{-cM}, \qquad \text{for } j \ne 1.$$

The features $\widetilde{T}_j(\boldsymbol{x})$ and $\widetilde{g}(\boldsymbol{c}_j)$ approximate the target affine expressions with bounded errors

$$|\widetilde{T}_j(\boldsymbol{x}) - T_j(\boldsymbol{x})| \le 3(P-1)M_g d\, e^{-cM},$$
$$|\widetilde{g}(\boldsymbol{c}_j) - g(\boldsymbol{c}_j)| \le (P-1)Be^{-cM}.$$

Next, applying the FFN layer $\mathcal{F}_1$ from Lemma 6 restores the structural components, yielding $\boldsymbol{Z}_1 = \mathcal{F}_1(\widehat{\boldsymbol{Z}}_1)$ with columns

$$(\boldsymbol{Z}_1)_{:,j} = \left[\widetilde{T}_j(\boldsymbol{x}), \widetilde{g}(\boldsymbol{c}_j), \mathbf{0}_{d-1}^\top, \delta_{j,1}, \sin(\theta_j), \cos(\theta_j)\right]^\top, \quad \forall j \in [P].$$

For Encoder Block 2, applying the MHA layer $\mathcal{A}_2$ from Lemma 7 that computes attention scores over $\widetilde{T}_j(\boldsymbol{x})$ and aggregates local function approximations $\widetilde{g}(\boldsymbol{c}_j)$. The output $\widehat{\boldsymbol{Z}}_2 = \mathcal{A}_2(\boldsymbol{Z}_1)$ has its first column to be

$$(\widehat{\boldsymbol{Z}}_2)_{:,1} = \left[\widehat{g}_T(\boldsymbol{x}), \mathbf{0}_{D-1}^\top\right]^\top = \widehat{g}_T(\boldsymbol{x})\boldsymbol{e}_1,$$

where

$$\widehat{g}_T(\boldsymbol{x}) = \sum_{j=1}^P \widetilde{\beta}_j(\boldsymbol{x})\widetilde{g}(\boldsymbol{c}_j), \quad \widetilde{\beta}_j(\boldsymbol{x}) = \frac{\exp(\widetilde{T}_j(\boldsymbol{x}))}{\sum_{l=1}^P \exp(\widetilde{T}_l(\boldsymbol{x}))},$$

all other columns ($k \ne 1$) evaluate to constant vectors $\widetilde{R}(\boldsymbol{x})\boldsymbol{e}_1$, where $\widetilde{R}(\boldsymbol{x}) = \frac{1}{P}\sum_{j=1}^P \widetilde{R}_j(\boldsymbol{x})$. The second point-wise FFN layer $\mathcal{F}_2$ preserves this output via an exact identity mapping, exploiting the ReLU property $x = \sigma(x) - \sigma(-x)$. Specifically, we set the hidden dimension $d_{\text{ff}} = 2D = 2d + 8$, and configure the weights ($\boldsymbol{W}_1 \in \mathbb{R}^{2D\times D}, \boldsymbol{W}_2 \in \mathbb{R}^{D\times 2D}$) and biases ($\boldsymbol{b}_1 \in \mathbb{R}^{2D}, \boldsymbol{b}_2 \in \mathbb{R}^D$) as

$$\boldsymbol{W}_1 = \begin{bmatrix} \boldsymbol{I}_D \\ -\boldsymbol{I}_D \end{bmatrix}, \quad \boldsymbol{b}_1 = \mathbf{0}_{2D}, \quad \boldsymbol{W}_2 = [\boldsymbol{I}_D \quad -\boldsymbol{I}_D], \quad \boldsymbol{b}_2 = \mathbf{0}_D.$$

This yields $\boldsymbol{Z}_2 = \mathcal{F}_2(\widehat{\boldsymbol{Z}}_2) = \boldsymbol{W}_2\sigma(\boldsymbol{W}_1\widehat{\boldsymbol{Z}}_2) = \widehat{\boldsymbol{Z}}_2$. Finally, the scalar regression output is then obtained by reading out the relevant components of $\boldsymbol{Z}_2$, via a linear affine mapping $\boldsymbol{c}_3^\top \mathrm{vec}(\boldsymbol{Z}_2)$, where setting $\boldsymbol{c}_3 = [1, 0, \ldots, 0]^\top \in \mathbb{R}^{DP}$ explicitly extracts $\widehat{g}_T(\boldsymbol{x})$.

Step 2: Error Bound Analysis. According to Lemma 8,

$$\sum_{j=1}^{P} |\widetilde{\beta}_j(\boldsymbol{x}) - \beta_j(\boldsymbol{x})| \le 2 \max_{j\in[P]} |\widetilde{T}_j(\boldsymbol{x}) - T_j(\boldsymbol{x})| \le 6(P-1)M_g d\, e^{-cM}.$$

It follows that the approximation error of the Transformer is bounded by

$$\begin{aligned}
|\widehat{g}_T(\boldsymbol{x}) - \widehat{g}(\boldsymbol{x})| &= \Big| \sum_{j=1}^{P} \widetilde{\beta}_j(\boldsymbol{x})\widetilde{g}(\boldsymbol{c}_j) - \sum_{j=1}^{P} \beta_j(\boldsymbol{x}) g(\boldsymbol{c}_j) \Big| \\
&\le \sum_{j=1}^{P} \widetilde{\beta}_j(\boldsymbol{x}) |\widetilde{g}(\boldsymbol{c}_j) - g(\boldsymbol{c}_j)| + \sum_{j=1}^{P} |\widetilde{\beta}_j(\boldsymbol{x}) - \beta_j(\boldsymbol{x})| |g(\boldsymbol{c}_j)| \\
&\le (P-1)Be^{-cM} + 6(P-1)M_g d\, e^{-cM} \cdot B \\
&= (P-1)B(1 + 6M_g d)e^{-cM}.
\end{aligned}$$

By setting the scaling factor $M = \frac{1}{c}\log\big(\frac{2(P-1)B(1+6M_g d)}{\varepsilon}\big)$, the implementation error is strictly bounded by $\varepsilon/2$. Combining this with the approximation error of the ideal Softmax POU, the triangle inequality yields

$$\sup_{\boldsymbol{x}\in[0,1]^d} |\widehat{g}_T(\boldsymbol{x}) - g(\boldsymbol{x})| \le \sup_{\boldsymbol{x}\in[0,1]^d} |\widehat{g}_T(\boldsymbol{x}) - \widehat{g}(\boldsymbol{x})| + \sup_{\boldsymbol{x}\in[0,1]^d} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Step 3: Parameter Complexity. We count the total architectural (dense) parameters $\mathcal{N}_{total}$ based on the embedding dimension $D = d + 4$:

- **Pre-processing**: Affine mapping $\mathbb{R}^d \to \mathbb{R}^{D\times P}$ has $D(P + d + 1)$ parameters.
- **MHA**$_1$: $H = P + 2$ heads with $d_k = d_v = 2$ have $8(P + 2)D$ parameters.
- **FFN**$_1$: Hidden dimension $2D$ has $4D^2 + 3D = 4d^2 + 35d + 76$ parameters.
- **MHA**$_2$: $H = 1$ head with $d_k = d_v = 1$ has $4D$ parameters.
- **FFN**$_2$: Hidden dimension $2D$ has $4D^2 + 3D = 4d^2 + 35d + 76$ parameters.
- **Readout**: Linear projection over $\mathrm{vec}(\boldsymbol{Z}_2)$ has DP parameters.

Summing these contributions yields:

$$\begin{aligned}
\mathcal{N}_{total} &= D(P + d + 1) + 8D(P + 2) + (4d^2 + 35d + 76) + 4D + (4d^2 + 35d + 76) + DP \\
&= 10P(d + 4) + 9d^2 + 95d + 236.
\end{aligned}$$

Since $P \le C_P\varepsilon^{-d/\alpha}$ and $\varepsilon \le 1/e \implies \varepsilon^{-d/\alpha} > 1$, we obtain

$$\mathcal{N}_{total} \le \big[10(d + 4)C_P + 9d^2 + 95d + 236\big]\varepsilon^{-d/\alpha} := C_N\varepsilon^{-d/\alpha}.$$

Step 4: Parameter Bounds. According to our construction, to bound the global parameter magnitude $M_{max} \le \max\big\{1, M, \frac{1}{1-2Pe^{-cM}}, (1 + (P-1)e^{-cM})B_{\mathcal{A}_1}\big\}$, we first bound M. Since $P \ge 2$, we have $\sin(\pi/P) \ge 2/P$, yielding

$$\frac{1}{c} = \frac{1}{2\sin^2(\pi/P)} \le \frac{P^2}{8} \le \frac{C_P^2}{8}\varepsilon^{-2d/\alpha}.$$

For $\varepsilon \le 1/e$, substituting $P \le C_P\varepsilon^{-d/\alpha}$ and $M_g \le C_M\varepsilon^{-2/\alpha}\log(2/\varepsilon)$ yields

$$\log\left(\frac{2(P-1)B(1+6M_g d)}{\varepsilon}\right) \le \log\left(\frac{2C_P B(1+6dC_M)}{\varepsilon^{(d+2+\alpha)/\alpha}}\log\frac{2}{\varepsilon}\right) \le C_{log}\log\frac{1}{\varepsilon},$$

where $C_{log} = \big|\log(4C_P B(1+6dC_M))\big| + \frac{d+2+\alpha}{\alpha} + 2$. It follows that

$$M = \frac{1}{c}\log\Big(\frac{4(P-1)B(1+6M_g d)}{\varepsilon}\Big) \le C_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, \quad \text{where } C_{mag} = \frac{C_P^2 C_{log}}{8}.$$

Next, by definition $e^{-cM} = \frac{\varepsilon}{2(P-1)B(1+6M_g d)}$. Since $1+6M_g d > 2 \ge \frac{P}{P-1}$, we have

$$2Pe^{-cM} = \frac{2P}{P-1}\frac{\varepsilon}{2B(1+6M_g d)} \le \frac{\varepsilon}{B} \implies \frac{1}{1-2Pe^{-cM}} \le \frac{1}{1-\frac{\varepsilon}{B}} \le \frac{1}{1-\frac{1}{eB}}.$$

Moreover, since $B_{\mathcal{A}_1} \le \max\{3M_g d, B, 1\} \le C_B\varepsilon^{-2/\alpha}\log(2/\varepsilon)$ with $C_B = \max\{3dC_M, B, 1\}$. Similarly bounding $(P-1)e^{-cM} \le \frac{\varepsilon}{2B}$, it follows that

$$(1+(P-1)e^{-cM})B_{\mathcal{A}_1} \le \Big(1+\frac{\varepsilon}{2B}\Big)C_B\varepsilon^{-2/\alpha}\log\frac{2}{\varepsilon} \le 2\Big(1+\frac{1}{2eB}\Big)C_B\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}.$$

Consequently, the global parameter magnitude satisfies

$$\begin{aligned} M_{max} &\le \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, 2\Big(1+\frac{1}{2eB}\Big)C_B\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}\right\} \\ &\le \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}, 2\Big(1+\frac{1}{2eB}\Big)C_B\right\}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon} := \widetilde{C}_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, \end{aligned} \tag{10}$$

This completes the proof. □

B.2 Approximation proof in Theorem 3

In this subsection, we prove Theorem 3 by adapting our constructive Transformer framework in Subsection B.1 to Riemannian manifolds.

Proof of Theorem 3. Applying Lemma 3 (Softmax POU Approximation on Manifolds) with target accuracy $\varepsilon/2$ yields $P = C_g \le C_P\varepsilon^{-d/\alpha}$ centers $\{\boldsymbol{c}_j\}_{j=1}^P \subset \mathcal{M}$, where $C_P = C_{\mathcal{M}}(16C_H)^{d/\alpha}$. Define the target affine features $T_j(\boldsymbol{x}) := 2M_g\boldsymbol{c}_j^\top\boldsymbol{x} - M_g\|\boldsymbol{c}_j\|_2^2$ and $R_j(\boldsymbol{x}) := g(\boldsymbol{c}_j)$. The corresponding Softmax POU approximation is given by

$$\widehat{g}(\boldsymbol{x}) = \sum_{j=1}^P \beta_j(\boldsymbol{x})g(\boldsymbol{c}_j) = \sum_{j=1}^P \frac{\exp(T_j(\boldsymbol{x}))}{\sum_{l=1}^P \exp(T_l(\boldsymbol{x}))}g(\boldsymbol{c}_j).$$

By Lemma 3, this achieves

$$\sup_{\boldsymbol{x}\in\mathcal{M}} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \le \varepsilon/2$$

provided that

$$M_g = C_M\varepsilon^{-2/\alpha}\log\frac{2}{\varepsilon}, \quad \text{where } C_M = \frac{(16C_H)^{2/\alpha}}{3}\left(\log(4BC_P) + \frac{d+\alpha}{\alpha}\right).$$

Since $\boldsymbol{x}, \boldsymbol{c}_j \in \mathcal{M} \subseteq [0,1]^{\bar{d}}$, the magnitudes of target features are bounded by $B_T := \max_{j\in[P]}\|T_j\|_\infty \le 3M_g\bar{d}$ and $B_R := \max_{j\in[P]}\|R_j\|_\infty \le B$.

Step 1: Transformer Construction. Following the exact architecture construction as in the proof of Theorem 1, but operating on the ambient dimension $\bar{d}$ instead of d, the pre-processing layer maps the input $\boldsymbol{x} \in \mathbb{R}^{\bar{d}}$ to $\boldsymbol{Z}_0 \in \mathbb{R}^{D\times P}$ using the embedding dimension $D = \bar{d}+4$. The first encoder block (MHA_1 and FFN_1) achieves local features and restores the structural components to exact canonical forms, yielding $\boldsymbol{Z}_1 \in \mathbb{R}^{D\times P}$ with columns

$$(\boldsymbol{Z}_1)_{:,j} = [\widetilde{T}_j(\boldsymbol{x}), \widetilde{g}(\boldsymbol{c}_j), \boldsymbol{0}_{\bar{d}-1}^\top, \delta_{j,1}, \sin(\theta_j), \cos(\theta_j)]^\top, \quad \forall j \in [P].$$

The local features approximate the targets with bounded errors

$$\begin{aligned} |\widetilde{T}_j(\boldsymbol{x}) - T_j(\boldsymbol{x})| &\le 3(P-1)M_g\bar{d}e^{-cM}, \\ |\widetilde{g}(\boldsymbol{c}_j) - g(\boldsymbol{c}_j)| &\le (P-1)Be^{-cM}. \end{aligned}$$

The second encoder block (MHA_2 and FFN_2) computes the softmax attention and aggregates the values, and the readout extracts

$$\widehat{g}_T(\boldsymbol{x}) = \sum_{j=1}^{P} \widetilde{\beta}_j(\boldsymbol{x})\widetilde{g}(\boldsymbol{c}_j), \quad \widetilde{\beta}_j(\boldsymbol{x}) = \frac{\exp(\widetilde{T}_j(\boldsymbol{x}))}{\sum_{l=1}^{P}\exp(\widetilde{T}_l(\boldsymbol{x}))}.$$

Step 2: Error Bound Analysis. By the softmax Lipschitz property Lemma 8,

$$\sum_{j=1}^{P} |\widetilde{\beta}_j(\boldsymbol{x}) - \beta_j(\boldsymbol{x})| \le 2\max_j |\widetilde{T}_j(\boldsymbol{x}) - T_j(\boldsymbol{x})| \le 6(P-1)M_g\bar{d}e^{-cM}.$$

The network approximation error is bounded by

$$\begin{aligned}
|\widehat{g}_T(\boldsymbol{x}) - \widehat{g}(\boldsymbol{x})| &\le \sum_{j=1}^{P} \widetilde{\beta}_j(\boldsymbol{x})|\widetilde{g}(\boldsymbol{c}_j) - g(\boldsymbol{c}_j)| + \sum_{j=1}^{P} |\widetilde{\beta}_j(\boldsymbol{x}) - \beta_j(\boldsymbol{x})||g(\boldsymbol{c}_j)| \\
&\le (P-1)Be^{-cM} + 6(P-1)M_g\bar{d}e^{-cM}B \\
&= (P-1)B(1+6M_g\bar{d})e^{-cM}.
\end{aligned}$$

Setting the MHA scaling factor $M = \frac{1}{c}\log\left(\frac{2(P-1)B(1+6M_g\bar{d})}{\varepsilon}\right)$ bounds this implementation error by $\varepsilon/2$. The triangle inequality yields the target bound

$$\sup_{\boldsymbol{x}\in\mathcal{M}} |\widehat{g}_T(\boldsymbol{x}) - g(\boldsymbol{x})| \le \sup_{\boldsymbol{x}\in\mathcal{M}} |\widehat{g}_T(\boldsymbol{x}) - \widehat{g}(\boldsymbol{x})| + \sup_{\boldsymbol{x}\in\mathcal{M}} |\widehat{g}(\boldsymbol{x}) - g(\boldsymbol{x})| \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Step 3: Parameter Complexity. With the embedding dimension $D = \bar{d} + 4$, similarly as the Proof of Theorem 1, the total number of dense architectural parameters $\mathcal{N}_{total}$ is

$$\begin{aligned}
\mathcal{N}_{total} &= D(P + \bar{d} + 1) + 8D(P+2) + (4\bar{d}^2 + 35\bar{d} + 76) + 4D \\
&\quad + (4\bar{d}^2 + 35\bar{d} + 76) + DP \\
&= 10P(\bar{d}+4) + 9\bar{d}^2 + 95\bar{d} + 236.
\end{aligned}$$

Since $P \le C_P\varepsilon^{-d/\alpha}$ and $\varepsilon \le 1/e \implies \varepsilon^{-d/\alpha} > 1$, we obtain

$$\mathcal{N}_{total} \le \left[10(\bar{d}+4)C_P + 9\bar{d}^2 + 95\bar{d} + 236\right]\varepsilon^{-d/\alpha} := C_N\varepsilon^{-d/\alpha}.$$

Notice that the parameter complexity scales strictly with $\varepsilon^{-d/\alpha}$, successfully bypassing the ambient dimension $\bar{d}$ in the exponent.

Step 4: Parameter Bounds. To bound the global parameter magnitude that $M_{max} \le \max\left\{1, M, \frac{1}{1-2Pe^{-cM}}, (1+(P-1)e^{-cM})B_{\mathcal{A}_1}\right\}$, we explicitly substitute $P \le C_P\varepsilon^{-d/\alpha}$ and $M_g \le C_M\varepsilon^{-2/\alpha}\log(2/\varepsilon)$ for $\varepsilon \le 1/e$. First, we bound M, since

$$\frac{1}{c} = \frac{1}{2\sin^2(\pi/P)} \le \frac{P^2}{8} \le \frac{C_P^2}{8}\varepsilon^{-2d/\alpha},$$

and

$$\begin{aligned}
\log\left(\frac{2(P-1)B(1+6M_g\bar{d})}{\varepsilon}\right) &\le \log\left(\frac{2C_PB\varepsilon^{-d/\alpha}\left(1+6\bar{d}C_M\varepsilon^{-2/\alpha}\log(2/\varepsilon)\right)}{\varepsilon}\right) \\
&\le \log\left(\frac{2C_PB(1+6\bar{d}C_M)}{\varepsilon^{(d+2+\alpha)/\alpha}}\log\frac{2}{\varepsilon}\right) \le C_{log}\log\frac{1}{\varepsilon},
\end{aligned}$$

where $C_{log} = \left|\log(2C_PB(1+6\bar{d}C_M))\right| + \frac{d+2+\alpha}{\alpha} + 2$. It follows that

$$M = \frac{1}{c}\log\left(\frac{2(P-1)B(1+6M_g\bar{d})}{\varepsilon}\right) \le C_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, \quad \text{where } C_{mag} = \frac{C_P^2C_{log}}{8}.$$

Next, by definition $e^{-cM} = \frac{\varepsilon}{2(P-1)B(1+6M_g\bar{d})}$. Since $1 + 6M_g d > 2 \geq \frac{P}{P-1}$, we bound the denominator term purely in ε:

$$2Pe^{-cM} = \frac{2P}{P-1}\frac{\varepsilon}{2B(1+6M_g\bar{d})} \leq \frac{\varepsilon}{B} \implies \frac{1}{1-2Pe^{-cM}} \leq \frac{1}{1-\frac{1}{eB}}.$$

For the feature magnitude, substituting M_g yields $B_{\mathcal{A}_1} \leq \max\{3M_g\bar{d}, B, 1\} \leq C_B\varepsilon^{-2/\alpha}\log(2/\varepsilon)$ with $C_B = \max\{3\bar{d}C_M, B, 1\}$. Similarly bounding $(P-1)e^{-cM} \leq \frac{\varepsilon}{2B}$, it follows that

$$(1+(P-1)e^{-cM})B_{\mathcal{A}_1} \leq \Big(1+\frac{\varepsilon}{2B}\Big)C_B\varepsilon^{-2/\alpha}\log\frac{2}{\varepsilon} \leq 2\Big(1+\frac{1}{2eB}\Big)C_B\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}.$$

Consequently, the global parameter magnitude satisfies

$$\begin{aligned} M_{max} &\leq \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, 2\Big(1+\frac{1}{2eB}\Big)C_B\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}\right\} \\ &\leq \max\left\{\frac{1}{1-\frac{1}{eB}}, C_{mag}, 2\Big(1+\frac{1}{2eB}\Big)C_B\right\}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon} := \widetilde{C}_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}, \end{aligned} \tag{11}$$

This completes the proof. □

C Proof of generalization results in Theorems 2 and 4

This final section details the proofs for the statistical generalization bounds of the empirical risk minimizer. We calculate the capacity of the hypothesis space and leverage it to derive near minimax-optimal learning rates.

C.1 Bounding the covering number

We first rigorously bound the covering number of our constructed Transformer hypothesis space in the following lemma. This provides the essential capacity measure necessary for controlling the statistical estimation error.

Lemma 9 (Covering Number of the Transformer Class)**.** *Let $\mathcal{T}$ be the class of Transformer networks defined in Theorem 1 (Theorem 3) with $L = 2$, sequence length P, embedding dimension $D = d + 4$ ($D = \bar{d} + 4$), and parameter magnitude bounded by $M_{max} \geq 1$. For any $\eta \in (0, 1]$, the covering number of $\mathcal{T}$ satisfies*

$$\log\mathcal{N}(\eta, \mathcal{T}, \|\cdot\|_\infty) \leq \mathcal{N}_{total}\log\left(\frac{1224256P^4D^{22}M_{max}^{26}}{\eta}\right),$$

where $\mathcal{N}_{total}$ is the total number of parameters.

Proof of Lemma 9. In the proof, we consider the class of Transformer networks defined in Theorem 3, the one defined in Theorem 1 follows the same result. Let $\widehat{g}_T(\boldsymbol{x};\boldsymbol{\theta})$ and $\widetilde{\widehat{g}}_T(\boldsymbol{x};\widetilde{\boldsymbol{\theta}})$ be two Transformer networks in $\mathcal{T}$ parameterized by $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ respectively, where $\|\boldsymbol{\theta}\|_\infty, \|\widetilde{\boldsymbol{\theta}}\|_\infty \leq M_{max}$. Assume the parameter difference is bounded by $\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\|_\infty \leq \delta$. We aim to bound the output difference $\|\widehat{g}_T - \widetilde{\widehat{g}}_T\|_\infty$ with respect to δ.

Step 1: Pre-processing and First Encoder Block. For any input $\boldsymbol{x} \in \mathcal{M} \subseteq [0,1]^{\bar{d}}$, the pre-processing layer maps $\boldsymbol{x}$ to the initial embedding matrix $\boldsymbol{Z}_0 \in \mathbb{R}^{D\times P}$ via

$$\boldsymbol{Z}_0 = \mathcal{P}(\boldsymbol{x}) = (\boldsymbol{W}_E\boldsymbol{x} + \boldsymbol{b}_E)\boldsymbol{e}_1^\top + \boldsymbol{P},$$

where $\boldsymbol{W}_E \in \mathbb{R}^{D\times\bar{d}}$, $\boldsymbol{b}_E \in \mathbb{R}^D$, and $\boldsymbol{P} \in \mathbb{R}^{D\times P}$ are parameterized by $\boldsymbol{\theta}$. Since $D = \bar{d} + 4$, $\|\boldsymbol{x}\|_\infty \leq 1$, and all trainable parameters are bounded by M_{max}, we calculate the bound for the magnitude $r_0 := \|\boldsymbol{Z}_0\|_{max}$ and the difference $\|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}$ as

$$\begin{aligned} r_0 &= \|(\boldsymbol{W}_E\boldsymbol{x} + \boldsymbol{b}_E)\boldsymbol{e}_1^\top + \boldsymbol{P}\|_{max} \leq \bar{d}\|\boldsymbol{W}_E\|_{max}\|\boldsymbol{x}\|_\infty + \|\boldsymbol{b}_E\|_{max} + \|\boldsymbol{P}\|_{max} \leq DM_{max}, \\ \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} &= \|((\boldsymbol{W}_E - \widetilde{\boldsymbol{W}}_E)\boldsymbol{x} + (\boldsymbol{b}_E - \widetilde{\boldsymbol{b}}_E))\boldsymbol{e}_1^\top + (\boldsymbol{P} - \widetilde{\boldsymbol{P}})\|_{max} \leq \bar{d}\delta + \delta + \delta \leq D\delta. \end{aligned}$$

Let $\mathcal{A}_1$ and $\mathcal{F}_1$ denote the MHA and FFN mappings in the first block. We first consider the concatenated output of the H_1 attention heads before the final projection, denoted as $\mathcal{H}_1(\boldsymbol{Z}_0) \in \mathbb{R}^{(H_1 d_v)\times P}$. For a single attention head $h \in [H_1]$, let $\boldsymbol{E}_1^h = \boldsymbol{Z}_0^\top (\boldsymbol{K}_1^h)^\top \boldsymbol{Q}_1^h \boldsymbol{Z}_0 \in \mathbb{R}^{P\times P}$ be the pre-softmax score matrix, $\boldsymbol{A}_1^h = \mathrm{Softmax}(\boldsymbol{E}_1^h) \in \mathbb{R}^{P\times P}$ be the attention probability matrix, and $\mathrm{head}_1^h = \boldsymbol{V}_1^h \boldsymbol{Z}_0 \boldsymbol{A}_1^h \in \mathbb{R}^{d_v \times P}$ be the head output. By Theorem 3, the inner dimensions are $d_k = d_v = 2$. We decouple the difference into a parameter error and an input error

$$\|\mathcal{H}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} \le \underbrace{\|\mathcal{H}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\boldsymbol{Z}_0)\|_{max}}_{\text{(I) Parameter Error}} + \underbrace{\|\widetilde{\mathcal{H}}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max}}_{\text{(II) Input Error}}.$$

(I) Parameter Error: Fixing the input $\boldsymbol{Z}_0$, for the pre-softmax matrix $\boldsymbol{E}_1^h$, we have

$$\begin{aligned}
\|(\boldsymbol{K}_1^h)^\top \boldsymbol{Q}_1^h - (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h\|_{max} &\le d_k \|\boldsymbol{K}_1^h\|_{max} \|\boldsymbol{Q}_1^h - \widetilde{\boldsymbol{Q}}_1^h\|_{max} + d_k \|\boldsymbol{K}_1^h - \widetilde{\boldsymbol{K}}_1^h\|_{max} \|\widetilde{\boldsymbol{Q}}_1^h\|_{max} \\
&\le 2 d_k M_{max} \delta \le 4 M_{max} \delta, \\
\|\boldsymbol{E}_1^h - \widetilde{\boldsymbol{E}}_1^h\|_{max} &= \|\boldsymbol{Z}_0^\top ((\boldsymbol{K}_1^h)^\top \boldsymbol{Q}_1^h - (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h) \boldsymbol{Z}_0\|_{max} \\
&\le D^2 \|\boldsymbol{Z}_0\|_{max}^2 \|(\boldsymbol{K}_1^h)^\top \boldsymbol{Q}_1^h - (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h\|_{max} \le 4 D^2 r_0^2 M_{max} \delta.
\end{aligned}$$

By Lemma 8, for each column $j \in [P]$,

$$\|(\boldsymbol{A}_1^h)_{:,j} - (\widetilde{\boldsymbol{A}}_1^h)_{:,j}\|_1 \le 2 \|(\boldsymbol{E}_1^h)_{:,j} - (\widetilde{\boldsymbol{E}}_1^h)_{:,j}\|_\infty \le 2 \|\boldsymbol{E}_1^h - \widetilde{\boldsymbol{E}}_1^h\|_{max} \le 8 D^2 r_0^2 M_{max} \delta.$$

Since $\|(\boldsymbol{A}_1^h)_{:,j}\|_1 = 1$, $\|\boldsymbol{Z}_0 \boldsymbol{A}_1^h\|_{max} \le \|\boldsymbol{Z}_0\|_{max} \le r_0$, it follows that

$$\begin{aligned}
&\|\mathrm{head}_1^h - \widetilde{\mathrm{head}}_1^h\|_{max} \\
&= \|(\boldsymbol{V}_1^h - \widetilde{\boldsymbol{V}}_1^h) \boldsymbol{Z}_0 \boldsymbol{A}_1^h + \widetilde{\boldsymbol{V}}_1^h \boldsymbol{Z}_0 (\boldsymbol{A}_1^h - \widetilde{\boldsymbol{A}}_1^h)\|_{max} \\
&\le D \|\boldsymbol{V}_1^h - \widetilde{\boldsymbol{V}}_1^h\|_{max} \|\boldsymbol{Z}_0 \boldsymbol{A}_1^h\|_{max} + \|\widetilde{\boldsymbol{V}}_1^h \boldsymbol{Z}_0\|_{max} \max_{j\in[P]} \|(\boldsymbol{A}_1^h)_{:,j} - (\widetilde{\boldsymbol{A}}_1^h)_{:,j}\|_1 \\
&\le D\delta r_0 + (D M_{max} r_0)(8 D^2 r_0^2 M_{max} \delta) \\
&\le 9 D^3 M_{max}^2 r_0^3 \delta.
\end{aligned}$$

Since $\mathcal{H}_1(\boldsymbol{Z}_0)$ concatenates all H_1 heads, we have

$$(I) \le 9 D^3 M_{max}^2 r_0^3 \delta.$$

(II) Input Error: Fixing parameters $\widetilde{\boldsymbol{\theta}}$, We have

$$\begin{aligned}
&\|\boldsymbol{Z}_0^\top (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h \boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0^\top (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h \widetilde{\boldsymbol{Z}}_0\|_{max} \\
&\le \|(\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0)^\top (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h \boldsymbol{Z}_0\|_{max} + \|\widetilde{\boldsymbol{Z}}_0^\top (\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h (\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0)\|_{max} \\
&\le 2D \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} \|(\widetilde{\boldsymbol{K}}_1^h)^\top \widetilde{\boldsymbol{Q}}_1^h \boldsymbol{Z}_0\|_{max} \\
&\le 2D \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} (D d_k M_{max}^2 r_0) \le 4 D^2 M_{max}^2 r_0 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}.
\end{aligned}$$

Notice that $\max_j \|(\boldsymbol{A}_1^h)_{:,j} - (\widetilde{\boldsymbol{A}}_1^h)_{:,j}\|_1 \le 8 D^2 M_{max}^2 r_0 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}$. It follows that

$$\begin{aligned}
&\|\widetilde{\boldsymbol{V}}_1^h \boldsymbol{Z}_0 \boldsymbol{A}_1^h - \widetilde{\boldsymbol{V}}_1^h \widetilde{\boldsymbol{Z}}_0 \widetilde{\boldsymbol{A}}_1^h\|_{max} \\
&= \|\widetilde{\boldsymbol{V}}_1^h (\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0) \boldsymbol{A}_1^h + \widetilde{\boldsymbol{V}}_1^h \widetilde{\boldsymbol{Z}}_0 (\boldsymbol{A}_1^h - \widetilde{\boldsymbol{A}}_1^h)\|_{max} \\
&\le D \|\widetilde{\boldsymbol{V}}_1^h\|_{max} \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} + \|\widetilde{\boldsymbol{V}}_1^h \widetilde{\boldsymbol{Z}}_0\|_{max} \max_j \|(\boldsymbol{A}_1^h)_{:,j} - (\widetilde{\boldsymbol{A}}_1^h)_{:,j}\|_1 \\
&\le D M_{max} \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} + (D M_{max} r_0)(8 D^2 M_{max}^2 r_0 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}) \\
&\le 9 D^3 M_{max}^3 r_0^2 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}.
\end{aligned}$$

Thus, we have

$$(II) \le 9 D^3 M_{max}^3 r_0^2 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max}.$$

Combining (I) and (II) with $r_0 \le DM_{max}$ and $\|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} \le D\delta$, we obtain

$$\begin{aligned}\|\mathcal{H}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} &\le 9M_{max}^2 D^3 r_0^3 \delta + 9M_{max}^3 D^3 r_0^2 \|\boldsymbol{Z}_0 - \widetilde{\boldsymbol{Z}}_0\|_{max} \\ &\le 9M_{max}^2 D^3 (DM_{max})^3 \delta + 9M_{max}^3 D^3 (DM_{max})^2 (D\delta) \\ &\le 18 D^6 M_{max}^5 \delta.\end{aligned}$$

Also, $\|\mathcal{H}_1(\boldsymbol{Z}_0)\|_{max} \le DM_{max} r_0 \le D^2 M_{max}^2$.

For $\mathcal{A}_1(\boldsymbol{Z}_0) = \boldsymbol{W}_1^O \mathcal{H}_1(\boldsymbol{Z}_0)$, since $H_1 d_v = (P+2) \times 2 \le 6P$, we have

$$\begin{aligned}\|\mathcal{A}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{A}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} &\le \|\boldsymbol{W}_1^O(\mathcal{H}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0))\|_{max} + \|(\boldsymbol{W}_1^O - \widetilde{\boldsymbol{W}}_1^O)\widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} \\ &\le (6P) M_{max} \|\mathcal{H}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} + (6P)\delta \|\widetilde{\mathcal{H}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} \\ &\le 6PM_{max}(18D^6 M_{max}^5 \delta) + 6P\delta(D^2 M_{max}^2) \\ &\le 114 P D^6 M_{max}^6 \delta,\end{aligned}$$

and $\|\mathcal{A}_1(\boldsymbol{Z}_0)\|_{max} \le (6P) M_{max} \|\mathcal{H}_1(\boldsymbol{Z}_0)\|_{max} \le 6PD^2 M_{max}^3$.

For the FFN layer, let $\boldsymbol{Y}_1 = \sigma(\boldsymbol{W}_1^1 \mathcal{A}_1(\boldsymbol{Z}_0) + \boldsymbol{b}_1^1)$ and $\widetilde{\boldsymbol{Y}}_1 = \sigma(\widetilde{\boldsymbol{W}}_1^1 \widetilde{\mathcal{A}}_1(\widetilde{\boldsymbol{Z}}_0) + \widetilde{\boldsymbol{b}}_1^1)$. Since σ is 1-Lipschitz, we have

$$\begin{aligned}\|\boldsymbol{Y}_1 - \widetilde{\boldsymbol{Y}}_1\|_{max} &\le \|(\boldsymbol{W}_1^1 - \widetilde{\boldsymbol{W}}_1^1)\mathcal{A}_1(\boldsymbol{Z}_0) + \widetilde{\boldsymbol{W}}_1^1(\mathcal{A}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{A}}_1(\widetilde{\boldsymbol{Z}}_0)) + (\boldsymbol{b}_1^1 - \widetilde{\boldsymbol{b}}_1^1)\|_{max} \\ &\le D\delta \|\mathcal{A}_1(\boldsymbol{Z}_0)\|_{max} + DM_{max} \|\mathcal{A}_1(\boldsymbol{Z}_0) - \widetilde{\mathcal{A}}_1(\widetilde{\boldsymbol{Z}}_0)\|_{max} + \delta \\ &\le D\delta(6PD^2 M_{max}^3) + DM_{max}(114 P D^6 M_{max}^6 \delta) + \delta \\ &\le 121 P D^7 M_{max}^7 \delta.\end{aligned}$$

The magnitude is bounded by

$$\|\boldsymbol{Y}_1\|_{max} \le DM_{max} \|\mathcal{A}_1(\boldsymbol{Z}_0)\|_{max} + M_{max} \le DM_{max}(6PD^2 M_{max}^3) + M_{max} \le 7PD^3 M_{max}^4.$$

Let $\boldsymbol{Z}_1 = \mathcal{F}_1(\mathcal{A}_1(\boldsymbol{Z}_0))$ and $\widetilde{\boldsymbol{Z}}_1 = \widetilde{\mathcal{F}}_1(\widetilde{\mathcal{A}}_1(\widetilde{\boldsymbol{Z}}_0))$. Since $d_{\text{ff}}^1 \le 2D$, it follows that the output difference of the first encoder block is bounded by

$$\begin{aligned}\|\boldsymbol{Z}_1 - \widetilde{\boldsymbol{Z}}_1\|_{max} &= \|(\boldsymbol{W}_1^2 \boldsymbol{Y}_1 + \boldsymbol{b}_1^2) - (\widetilde{\boldsymbol{W}}_1^2 \widetilde{\boldsymbol{Y}}_1 + \widetilde{\boldsymbol{b}}_1^2)\|_{max} \\ &\le \|(\boldsymbol{W}_1^2 - \widetilde{\boldsymbol{W}}_1^2)\boldsymbol{Y}_1 + \widetilde{\boldsymbol{W}}_1^2(\boldsymbol{Y}_1 - \widetilde{\boldsymbol{Y}}_1) + (\boldsymbol{b}_1^2 - \widetilde{\boldsymbol{b}}_1^2)\|_{max} \\ &\le d_{\text{ff}}^1 \delta(7PD^3 M_{max}^4) + d_{\text{ff}}^1 M_{max}(121 P D^7 M_{max}^7 \delta) + \delta \\ &\le 257 P D^8 M_{max}^8 \delta.\end{aligned}$$

Its magnitude is bounded by

$$\begin{aligned}r_1 := \|\boldsymbol{Z}_1\|_{max} &\le d_{\text{ff}}^1 \|\boldsymbol{W}_1^2\|_{max} \|\boldsymbol{Y}\|_{max} + \|\boldsymbol{b}_1^2\|_{max} \\ &\le 2DM_{max}(7PD^3 M_{max}^4) + M_{max} \\ &\le 15 P D^4 M_{max}^5.\end{aligned}$$

Step 2: Second Encoder Block. We recursively apply the same logic to the second block, utilizing $\boldsymbol{Z}_1$ as the input bounded by r_1. For the second block, since $H_2 = 1$ and the inner dimensions are $d_k^2 = d_v^2 = 1$, the MHA bound simplifies. Adapting the derivation from Step 1 by substituting r_0 with r_1, $\boldsymbol{Z}_0$ with $\boldsymbol{Z}_1$, and updating the constants for $d_k = 1$, we have

$$\begin{aligned}&\|\mathcal{H}_2(\boldsymbol{Z}_1) - \widetilde{\mathcal{H}}_2(\widetilde{\boldsymbol{Z}}_1)\|_{max} \\ &\le 5M_{max}^2 D^3 r_1^3 \delta + 5M_{max}^3 D^3 r_1^2 \|\boldsymbol{Z}_1 - \widetilde{\boldsymbol{Z}}_1\|_{max} \\ &\le 5M_{max}^2 D^3 (15 P D^4 M_{max}^5)^3 \delta + 5M_{max}^3 D^3 (15 P D^4 M_{max}^5)^2 (257 P D^8 M_{max}^8 \delta) \\ &\le 16875 P^3 D^{15} M_{max}^{17} \delta + 289125 P^3 D^{19} M_{max}^{21} \delta \\ &\le 306000 P^3 D^{19} M_{max}^{21} \delta.\end{aligned}$$

Its magnitude is bounded by $\|\mathcal{H}_2(\boldsymbol{Z}_1)\|_{max} \le DM_{max} r_1 \le 15 P D^5 M_{max}^6$.

Applying the output projection $\boldsymbol{W}_2^O \in \mathbb{R}^{D\times 1}$ (since $H_2 d_v^2 = 1$), we have

$$\begin{aligned}\|\mathcal{A}_2(\boldsymbol{Z}_1) - \widetilde{\mathcal{A}}_2(\widetilde{\boldsymbol{Z}}_1)\|_{max} &\le 1 \cdot M_{max}\|\mathcal{H}_2(\boldsymbol{Z}_1) - \widetilde{\mathcal{H}}_2(\widetilde{\boldsymbol{Z}}_1)\|_{max} + 1 \cdot \delta\|\widetilde{\mathcal{H}}_2(\widetilde{\boldsymbol{Z}}_1)\|_{max} \\ &\le M_{max}(306000P^3D^{19}M_{max}^{21}\delta) + \delta(15PD^5M_{max}^6) \\ &\le 306015P^3D^{19}M_{max}^{22}\delta.\end{aligned}$$

The magnitude is $\|\mathcal{A}_2(\boldsymbol{Z}_1)\|_{max} \le M_{max}\|\mathcal{H}_2(\boldsymbol{Z}_1)\|_{max} \le 15PD^5M_{max}^7$.

For the FFN layer, we define $\boldsymbol{Y}_2 = \sigma(\boldsymbol{W}_2^1\mathcal{A}_2(\boldsymbol{Z}_1) + \boldsymbol{b}_2^1)$ and $\widetilde{\boldsymbol{Y}}_2 = \sigma(\widetilde{\boldsymbol{W}}_2^1\widetilde{\mathcal{A}}_2(\widetilde{\boldsymbol{Z}}_1) + \widetilde{\boldsymbol{b}}_2^1)$, we have

$$\begin{aligned}\|\boldsymbol{Y}_2\|_{max} &\le DM_{max}\|\mathcal{A}_2(\boldsymbol{Z}_1)\|_{max} + M_{max} \le 16PD^6M_{max}^8, \\ \|\boldsymbol{Y}_2 - \widetilde{\boldsymbol{Y}}_2\|_{max} &\le D\delta\|\mathcal{A}_2(\boldsymbol{Z}_1)\|_{max} + DM_{max}\|\mathcal{A}_2(\boldsymbol{Z}_1) - \widetilde{\mathcal{A}}_2(\widetilde{\boldsymbol{Z}}_1)\|_{max} + \delta \\ &\le 306031P^3D^{20}M_{max}^{23}\delta.\end{aligned}$$

Let $\boldsymbol{Z}_2 = \mathcal{F}_2(\mathcal{A}_2(\boldsymbol{Z}_1))$ and $\widetilde{\boldsymbol{Z}}_2 = \widetilde{\mathcal{F}}_2(\widetilde{\mathcal{A}}_2(\widetilde{\boldsymbol{Z}}_1))$. Since $d_{\text{ff}}^2 \le 2D$, it follows that

$$\begin{aligned}\|\boldsymbol{Z}_2 - \widetilde{\boldsymbol{Z}}_2\|_{max} &\le 2D\delta\|\boldsymbol{Y}_2\|_{max} + 2DM_{max}\|\boldsymbol{Y}_2 - \widetilde{\boldsymbol{Y}}_2\|_{max} + \delta \\ &\le 2D\delta(16PD^6M_{max}^8) + 2DM_{max}(306031P^3D^{20}M_{max}^{23}\delta) + \delta \\ &\le 612095P^3D^{21}M_{max}^{24}\delta.\end{aligned}$$

The magnitude of the final hidden state is bounded by

$$r_2 := \|\boldsymbol{Z}_2\|_{max} \le 2DM_{max}\|\boldsymbol{Y}_2\|_{max} + M_{max} \le 33PD^7M_{max}^9.$$

Step 3: Readout Layer and Covering Number. The final output is computed by the linear readout vector $c \in \mathbb{R}^{PD}$, where $\|c\|_\infty \le M_{max}$. The network approximation difference is bounded by

$$\begin{aligned}\|\widehat{g}_T(\boldsymbol{x};\boldsymbol{\theta}) - \widetilde{\widehat{g}}_T(\boldsymbol{x};\widetilde{\boldsymbol{\theta}})\|_\infty &\le \|c\|_1\|\boldsymbol{Z}_2 - \widetilde{\boldsymbol{Z}}_2\|_{max} + \|c - \widetilde{c}\|_1\|\widetilde{\boldsymbol{Z}}_2\|_{max} \\ &\le (PDM_{max})(612095P^3D^{21}M_{max}^{24}\delta) + (PD\delta)(33PD^7M_{max}^9) \\ &\le 612128P^4D^{22}M_{max}^{25}\delta.\end{aligned}$$

This establishes the global Lipschitz constant $L_{param} = 612128P^4D^{22}M_{max}^{25}$ with respect to the parameters.

To obtain an η-covering of $\mathcal{T}$ in the function space with respect to the $\|\cdot\|_\infty$ norm, it suffices to construct a δ-covering of the parameter space $\Theta = [-M_{max}, M_{max}]^{\mathcal{N}_{total}}$, where $\delta = \eta/L_{param}$. The covering number of the hypercube Θ is bounded by $(2M_{max}/\delta)^{\mathcal{N}_{total}}$. Taking the logarithm yields

$$\log\mathcal{N}(\eta, \mathcal{T}, \|\cdot\|_\infty) \le \mathcal{N}_{total}\log\left(\frac{2M_{max}L_{param}}{\eta}\right) = \mathcal{N}_{total}\log\left(\frac{1224256P^4D^{22}M_{max}^{26}}{\eta}\right).$$

This completes the proof. □

C.2 Deriving learning rates of the ERM algorithm

To establish the generalization bound, we first introduce the following oracle inequality for the ERM algorithm defined over a compact subset of continuous functions.

Lemma 10 ([5], [27]). *Suppose there exist constants $n', \nu > 0$ such that the covering number of the hypothesis space $\mathcal{T}$ satisfies*

$$\log\mathcal{N}(\eta, \mathcal{T}, \|\cdot\|_\infty) \le n'\log\frac{\nu}{\eta}, \quad \forall\eta > 0.$$

Then for any $h^* \in \mathcal{T}$, and any $\eta > 0$, we have

$$\begin{aligned}Prob\left\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > \eta + 2\|h^* - f_\rho\|_\rho^2\right\} \le \exp\left\{n'\log\frac{16\nu B}{\eta} - \frac{3n\eta}{512B^2}\right\} \\ + \exp\left\{\frac{-3n\eta^2}{16(3B + \|h^*\|_\infty)^2(6\|h^* - f_\rho\|_\rho^2 + \eta)}\right\}.\end{aligned}$$

Utilizing the covering number bounds Lemma 9 established above, we complete the proofs for Theorem 2 and Theorem 4. We optimize the bias-variance trade-off via the oracle inequality Lemma 10 to achieve the final convergence rates.

Proof of Theorem 2. By Theorem 1, for any $\varepsilon \in (0, 1/e]$, there exists a Transformer $h \in \mathcal{T}$ such that $\|h - f_\rho\|_\infty \le \varepsilon$. Since $\|f_\rho\|_\infty \le B$, we have $\|h\|_\infty \le B + \varepsilon \le B + 1$. By Lemma 9, we have

$$\log \mathcal{N}(\eta, \mathcal{T}, \|\cdot\|_\infty) \le \mathcal{N}_{total} \log\left(\frac{1224256 P^4 D^{22} M_{max}^{26}}{\eta}\right).$$

Substituting $\mathcal{N}_{total} \le C_N \varepsilon^{-d/\alpha}$, $P \le C_P \varepsilon^{-d/\alpha}$, $D = d + 4$, and $M_{max} \le \widetilde{C}_{mag} \varepsilon^{-2d/\alpha} \log \frac{1}{\varepsilon}$, we set $n' = C_N \varepsilon^{-d/\alpha}$ and obtain

$$\nu = 1224256 (C_P \varepsilon^{-d/\alpha})^4 (d+4)^{22} \Big(\widetilde{C}_{mag} \varepsilon^{-2d/\alpha} \log \frac{1}{\varepsilon}\Big)^{26} \le C_\nu \varepsilon^{-56d/\alpha} \Big(\log \frac{1}{\varepsilon}\Big)^{26},$$

where $C_\nu = 1224256 C_P^4 (d+4)^{22} \widetilde{C}_{mag}^{26}$.

Applying Lemma 10 with $h^* = h$ and $\eta \ge 6\varepsilon^2$, we have

$$\begin{aligned}\text{Prob}\Big\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > 2\eta\Big\} \le \exp\left\{C_N \varepsilon^{-d/\alpha} \log \frac{16\nu B}{\eta} - \frac{3n\eta}{512 B^2}\right\} \\ + \exp\left\{\frac{-3n\eta^2}{16(4B+1)^2(6\varepsilon^2 + \eta)}\right\}.\end{aligned}$$

For $\eta \ge 6\varepsilon^2$, since $\varepsilon \le 1/e$, we have $\log \frac{16\nu B}{\eta} \le \big(\log \frac{8BC_\nu}{3} + 28 + \frac{56d}{\alpha}\big) \log \frac{1}{\varepsilon}$. If we further restrict

$$\eta \ge \max\left\{6\varepsilon^2, C_1 \frac{\varepsilon^{-d/\alpha} \log \frac{1}{\varepsilon}}{n}\right\}, \tag{12}$$

where $C_1 = \frac{1024 B^2 C_N}{3} \big(\log \frac{8BC_\nu}{3} + 28 + \frac{56d}{\alpha}\big)$, then we have

$$\begin{aligned}&\text{Prob}\Big\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > 2\eta\Big\} \\ \le\ & \exp\left\{C_N \varepsilon^{-d/\alpha} \log \frac{16\nu B}{\eta} - \frac{3n\eta}{512B^2}\right\} + \exp\left\{\frac{-3n\eta^2}{16(4B+1)^2(6\varepsilon^2+\eta)}\right\} \\ \le\ & \exp\left\{\frac{3n\eta}{1024B^2} - \frac{3n\eta}{512B^2}\right\} + \exp\left\{\frac{-3n\eta}{32(4B+1)^2}\right\} \\ \le\ & \exp\left\{-\frac{3n\eta}{1024B^2}\right\} + \exp\left\{-\frac{3n\eta}{32(4B+1)^2}\right\},\end{aligned}$$

where the second inequality follows from the restriction (12). Furthermore, by setting $t = 2\eta$ and defining

$$C_3 = \min\left\{\frac{3}{2048B^2}, \frac{3}{64(4B+1)^2}\right\},$$

we arrive at

$$\text{Prob}\Big\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > t\Big\} \le 2\exp(-C_3 n t), \qquad \forall t \ge C_2 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha} \log \frac{1}{\varepsilon}}{n}\right\}, \tag{13}$$

where $C_2 = \max\{12, 2C_1\}$. If we apply the formula for the mean of the non-negative random variable $\xi = \|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2$:

$$\mathbb{E}[\xi] = \int_0^\infty \text{Prob}[\xi > t] dt,$$

we see from (13) that with $\Delta := C_2 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\}$, there holds

$$\begin{aligned}\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] &= \left(\int_0^\Delta + \int_\Delta^\infty\right) \text{Prob}\Big\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > t\Big\}dt \\ &\le \Delta + \int_\Delta^\infty 2\exp(-C_3 nt)dt \\ &\le \Delta + \frac{2}{C_3 n} \le C_4 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\},\end{aligned} \tag{14}$$

where $C_4 = C_2 + \frac{2}{C_3}$. Finally, by choosing $\varepsilon = n^{-\frac{\alpha}{2\alpha+d}}$, we obtain

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] \le C_4 n^{-\frac{2\alpha}{2\alpha+d}}\log n,$$

Thus, we complete the proof. □

Proof of Theorem 4. The proof follows the same logic as the proof of Theorem 2, utilizing the covering number bound of $\mathcal{T}$ by Lemma 9, with the approximation complexities from Theorem 3 depending on the intrinsic dimension d:

$$\mathcal{N}_{total} \le C_N \varepsilon^{-d/\alpha}, \quad P \le C_P \varepsilon^{-d/\alpha}, \quad M_{max} \le \widetilde{C}_{mag}\varepsilon^{-2d/\alpha}\log\frac{1}{\varepsilon}.$$

If we restrict

$$\eta \ge \max\left\{6\varepsilon^2, C_1\frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\},$$

where $C_1 = \frac{1024B^2C_N}{3}\left(\log\frac{8BC_\nu}{3} + 28 + \frac{56d}{\alpha}\right)$, $C_\nu = 1224256C_P^4(d+4)^{22}\widetilde{C}_{mag}^{26}$, then we have

$$\text{Prob}\Big\{\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2 > 2\eta\Big\} \le \exp\left\{-\frac{3n\eta}{1024B^2}\right\} + \exp\left\{-\frac{3n\eta}{32(4B+1)^2}\right\}.$$

Furthermore, setting

$$\widetilde{C}_4 = \max\{12, 2C_1\} + 2\max\left\{\frac{2048B^2}{3}, \frac{64(4B+1)^2}{3}\right\}, \tag{15}$$

we arrive at

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] \le \widetilde{C}_4 \max\left\{\varepsilon^2, \frac{\varepsilon^{-d/\alpha}\log\frac{1}{\varepsilon}}{n}\right\}.$$

Finally, by choosing $\varepsilon = n^{-\frac{\alpha}{2\alpha+d}}$, we derive

$$\mathbb{E}\left[\|\pi_B f_{\mathcal{S}} - f_\rho\|_\rho^2\right] \le \widetilde{C}_4 n^{-\frac{2\alpha}{2\alpha+d}}(\log n),$$

Thus, we complete the proof. □

D Numerical experiment: details

D.1 Data, architecture, and training

Data. For each component $k = 1, \ldots, 8$, independently draw gaps $r_\ell \sim \text{Unif}[0.5, 1.5]$ ($\ell = 1, \ldots, 8$) and heights $h_\ell \sim \text{Unif}[-1, 1]$ ($\ell = 0, \ldots, 7$). Set $t_0 = 0$, $t_\ell = \sum_{m\le\ell} r_m / \sum_{m=1}^8 r_m$, and $h_8 = h_0$. The target is

$$g_k(\boldsymbol{x}(u)) = \frac{(t_{\ell+1} - u)h_\ell + (u - t_\ell)h_{\ell+1}}{t_{\ell+1} - t_\ell}, \quad t_\ell \le u < t_{\ell+1}. \tag{16}$$

Training inputs are sampled with $u \sim \text{Unif}[0, 1)$.

Architecture. Using Section 2's column-token convention and unscaled dot-product attention,

$$\begin{aligned} \boldsymbol{Z}^{(0)} &= [\boldsymbol{E}\boldsymbol{x} + \boldsymbol{b}_E, \mathbf{0}, \ldots, \mathbf{0}] + [\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{127}] \in \mathbb{R}^{16\times 128}, \\ \boldsymbol{U} &= \boldsymbol{Z}^{(0)} + \mathrm{MHA}_1(\boldsymbol{Z}^{(0)}), \qquad \boldsymbol{Z}^{(1)} = \boldsymbol{U} + \boldsymbol{W}_2\,\mathrm{ReLU}(\boldsymbol{W}_1\boldsymbol{U} + \boldsymbol{b}_1\mathbf{1}^\mathsf{T}) + \boldsymbol{b}_2\mathbf{1}^\mathsf{T}. \end{aligned} \tag{17}$$

The first block has two heads with $(d_k, d_v) = (2, 2)$ and FFN width 4; the second has one head with $(d_k, d_v) = (4, 8)$. The network has 3020 parameters. The training parameters use independent zero-mean Gaussian initialization with standard deviations:

Parameter	Std.	Parameter	Std.
$\boldsymbol{E},\ \boldsymbol{b}_E$	$1/\sqrt{2},\ 0.1$	Tokens $\boldsymbol{p}_j$	$1/4$
First/second $\boldsymbol{Q}, \boldsymbol{K}$	$1/4,\ 1/2$	Both value projections	$1/4$
First/second output projections	$1/2,\ 1/\sqrt{8}$	FFN weights/biases	$1/\sqrt{10},\ 0.05$
Readout weight/bias	$1/4,\ 0$		

Optimization. The squared loss is

$$\mathcal{L} = \frac{1}{2 \cdot 1024 \cdot 8} \sum_{i=1}^{1024} \|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{g}(\boldsymbol{x}_i)\|_2^2. \tag{18}$$

Adam uses 14000 updates, batches of 256 sampled with replacement, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 10^{-8}$, with cosine learning-rate decay from 0.003 to 0.0003. From the best recorded Adam state, full-batch L-BFGS uses 40 calls with `max_iter`$= 10$, `max_eval`$= 12$, step size 1, history 30, strong-Wolfe search, and gradient/change tolerances $10^{-8}/10^{-11}$. Training uses float32. The final checkpoint is selected as the state with the lowest training loss among the initialization and checkpoints recorded every 100 Adam updates and after each L-BFGS call. Calibration data do not select checkpoints. Implementation uses PyTorch 2.7.1+cu118 and NumPy 2.3.5 on a shared GTX 1070 (8 GB); recorded time for training and diagnostics was 269 seconds.

D.2 Evaluation and Figure 2

At the $N = 4096$ evaluation inputs $\boldsymbol{x}_i^{\mathrm{test}} = \boldsymbol{x}((i-1)/N)$, let $y_{ik} = g_k(\boldsymbol{x}_i^{\mathrm{test}})$ and $\widehat{y}_{ik} = h_k(\boldsymbol{x}_i^{\mathrm{test}})$ for a predictor $\boldsymbol{h}$. The NMSE is defined as the average of the variance-normalized mean squared prediction errors across the eight output components:

$$\mathrm{NMSE}(\boldsymbol{h}, \boldsymbol{g}) = \frac{1}{8}\sum_{k=1}^{8} \frac{\sum_{i=1}^{N}(\widehat{y}_{ik} - y_{ik})^2}{\sum_{i=1}^{N}(y_{ik} - \bar{y}_k)^2}, \qquad \bar{y}_k = \frac{1}{N}\sum_{i=1}^{N} y_{ik}. \tag{19}$$

To select active sources and order the heatmap rows, we sample 2048 calibration inputs independently of the training data, with $u \sim \mathrm{Unif}[0, 1)$. Trained attention on these inputs defines peak locations $\boldsymbol{c}_j$ and active sources $\mathcal{A}$:

$$\boldsymbol{c}_j = \boldsymbol{x}^{\mathrm{cal}}_{\arg\max_i a_j(\boldsymbol{x}_i^{\mathrm{cal}})}, \qquad \mathcal{A} = \left\{ j \in \mathcal{S} : \frac{1}{2048}\sum_i a_j(\boldsymbol{x}_i^{\mathrm{cal}}) \geq \frac{0.32}{|\mathcal{S}|} \right\}. \tag{20}$$

Figure 2(a,b,d) orders sources by the circle position of $\boldsymbol{c}_j$, using the same order for both states. Panel (c) plots the trained weights of the 25 active sources; predictions retain all 127 sources. Panel (d) selects the evaluation input with the smallest maximum initial attention weight, breaking ties by the smallest u, giving $u = 0.1553$.

D.3 Local target approximation

With initial or trained attention held fixed, we fit one affine function of $\boldsymbol{x}$ per source to the training targets and aggregate these functions using the same attention weights:

$$(\widehat{\boldsymbol{B}}_j, \widehat{\boldsymbol{b}}_j) \in \arg\min_{\boldsymbol{B},\boldsymbol{b}} \sum_{i=1}^{1024} a_j(\boldsymbol{x}_i)\|\boldsymbol{B}\boldsymbol{x}_i + \boldsymbol{b} - \boldsymbol{g}(\boldsymbol{x}_i)\|_2^2, \qquad \widetilde{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{j\in\mathcal{S}} a_j(\boldsymbol{x})(\widehat{\boldsymbol{B}}_j\boldsymbol{x} + \widehat{\boldsymbol{b}}_j). \tag{21}$$

Constant fits impose $\boldsymbol{B} = \mathbf{0}$. Minimum-norm weighted least squares uses float64 SVD with relative cutoff 10^{-10}. Only training data are used for these fits. Targets and predictions are evaluated on the 4096 equally spaced circle points specified in Appendix D.2.

Table 2: NMSE for the network in Figure 2. Local fits use (21) with initial or trained attention held fixed.

Predictor	Initial attention	Trained attention
Trained network	—	9.07×10^{-4}
Constant target fits	0.551	0.125
Affine target fits	0.165	0.00664

Table 2 examines how the spatial structure of attention affects the aggregation of source-wise constant and affine fits. At initialization, the relatively diffuse attention weights do not induce clearly localized fitting regions. After training, the attention weights become more spatially concentrated, allowing the same simple fitting functions to serve as effective local approximants. In particular, the NMSE of the affine reconstruction decreases from 0.165 with initial attention to 0.00664 with trained attention. This suggests that learned softmax attention can induce a useful spatial localization structure for local-to-global approximation, consistent with the underlying principle of our Softmax POU framework.

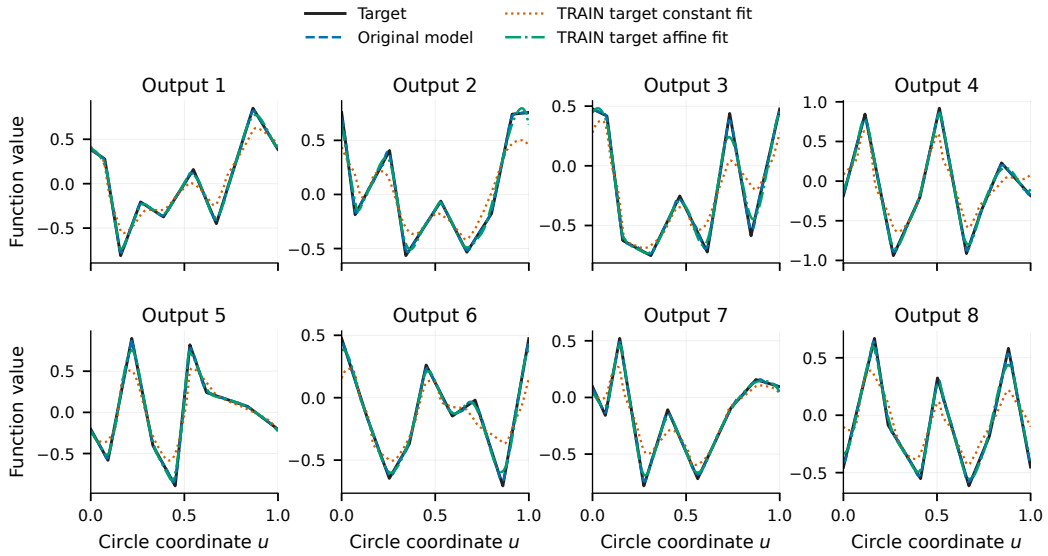


Figure 3: Eight regression outputs: targets, trained model predictions, and local constant/affine fits. Local fits use frozen trained attention.

Figure 3 provides a component-wise comparison of the target functions, the trained Transformer predictions, and the attention-weighted constant and affine reconstructions. The affine fits follow the target variations more closely than the constant fits, highlighting the complementary roles of local approximation and attention-based aggregation: affine functions capture variations within individual neighborhoods, while the learned attention weights smoothly combine their contributions across the circle. This behavior is qualitatively consistent with our Softmax POU framework, in which spatially localized weights assemble local approximants into a global predictor. Furthermore, the trained Transformer achieves even lower error than the affine reconstruction, suggesting that its learned source representations capture finer local variations than the simple affine approximants.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical results are proved in the Appendix. Section 5 presents a regression experiment examining learned attention localization.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 notes the restriction to Hölder exponents $\alpha \in (0, 1]$. The numerical study uses noiseless regression on the circle.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions (e.g., α-Hölder continuity, manifold reach $\tau > 0$) are explicitly stated in the Theorems and Lemmas within the main text. We provide proof sketches in the main text to build intuition, and the complete, rigorous mathematical proofs are deferred to the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 and Appendix D specify target generation, sample sizes, architecture, initialization, optimization, and evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experimental code and checkpoints have not been publicly released. Appendix D provides the experimental protocol and synthetic data definitions; a reproduction package has been prepared locally, but this does not constitute open access.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 and Appendix D give the data and architecture settings, random token initialization, optimizer and training budget, diagnostics, and checkpoint selection.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: No error bars, confidence intervals, or statistical significance tests are reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D.1 gives the GPU, software versions, optimization budget, and recorded time for training and diagnostics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research combines mathematical analysis with synthetic regression experiments. It involves no human subjects, personal or sensitive data, or hazardous applications, and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This is foundational research on approximation and generalization, illustrated using synthetic regression tasks. It is not tied to a particular real-world application or deployment, and the work does not identify a direct path to societal harm.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The experiments use small regression models and analytically defined synthetic data. They do not involve high-risk pretrained models, scraped datasets, or other assets requiring controlled-release safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The experiments use PyTorch and NumPy (BSD-style licenses, with their bundled dependency notices) and Matplotlib (the Matplotlib license agreement for versions 1.3.0 and later); training software versions are recorded in Appendix D. The data are generated from the stated analytic targets, and no third-party datasets or pretrained models are used.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not publicly release a new dataset, codebase, or pretrained model. The synthetic data construction and experimental protocol are documented in Appendix D.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects. All experimental data are synthetically generated.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research does not involve human subjects, and therefore Institutional Review Board (IRB) approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: No LLM is used as a component of the proposed Transformer architecture, theoretical construction, or optimization method. AI assistance with manuscript preparation and numerical experiment implementation, execution, analysis, and visualization is disclosed in the acknowledgments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.