NEURAL SCALING LAWS OF TRANSFORMER OPERATOR NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers have emerged as powerful architectures for learning solution operators of physical systems. Empirically the prediction error has been observed to decrease when the data size and model size increase, suggesting neural scaling behavior. Yet a theoretical understanding of such scaling laws for transformer-based operator learning remains limited. In this work, we develop a theoretical framework for characterizing the approximation and generalization errors of transformer-based operator learning. Our analysis builds on a local-to-global approximation principle that is naturally aligned with the softmax attention mechanism and yields discretization-invariant output functions. On approximation theory, we derive a universal approximation error of transformer-based operator learning for Hölder-regular operators. On generalization theory, we establish a power scaling law between the prediction error and the training data size. The rate of convergence represented by the scaling exponent explicitly reflects the dimensions of the input and output domains, the regularity of the underlying functions and operators, and crucially, the intrinsic dimension of the input function class. By exploiting this intrinsic low-dimensional structure, our analysis yields a power-law generalization rate for operator learning, in contrast to the logarithmic-type power-law rates appearing in existing analyses of operator learning with feedforward neural networks. Numerical experiments validate the predicted power-law scaling and confirm that the convergence rate varies systematically with the intrinsic dimension of the input function class.

1 INTRODUCTION

Learning solution operators for differential equations and operators arising in inverse problems is a fundamental problem in science and engineering. Many problems in applications can be formulated as an operator between function spaces where the input may represent an initial condition, boundary condition, forcing term, material coefficient, or geometric description, and the output represents the corresponding solution or quantity of interest. Operator-learning methods seek to learn the input-to-output map and can therefore be repeatedly evaluated for new inputs once training is completed. This perspective has motivated a rapidly growing family of neural operator architectures, including Deep Operator Networks (DeepONets) (Lu et al., 2021), Fourier Neural Operators (FNOs) (Li et al., 2020) and other neural operators, together with numerous subsequent developments in theory (Lanthaler et al., 2022; Kovachki et al., 2024; Liu et al., 2024; 2026; Yang et al., 2026) and computation (Pathak et al., 2022; Li et al., 2023).

More recently, Transformers have become increasingly prominent in operator learning. The attention mechanism in Transformers (Vaswani et al., 2017) provides a natural way to represent samples of an input function as tokens, aggregate information globally across these observations, and generate predictions at arbitrary output locations. A growing family of transformer-based operator architectures has demonstrated strong performance for learning solution operators of partial differential equations and other physical systems, including the Galerkin Transformer (Cao, 2021), OFormer (Li et al., 2022), GNOT (Hao et al., 2023), Position-induced Transformer (PiT) (Chen & Wu, 2024), Transolver (Wu et al., 2024), and many others (Bryutkin et al., 2024; Luo et al., 2024). Numerical experiments in these works (Li et al., 2022; Hao et al., 2023) have demonstrated systematic improvements as training data increase, suggesting neural scaling behavior, which appears in a wide range of AI models (Kaplan et al., 2020; Havrilla & Liao, 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These empirical advances motivate a quantitative theoretical understanding of how the approximation and generalization performance of transformer-based operator learning scales with model complexity, training-data size, and the underlying structure of the function spaces. Understanding these scaling laws can in turn guide the allocation of model capacity and training data, enabling more efficient use of computational resources.

When feedforward neural networks are utilized for operator learning, neural scaling laws about the approximation and generalization errors are theoretically characterized for DeepONets (Lanthaler et al., 2022; Liu et al., 2026), FNOs (Kovachki et al., 2021) and an encoder-decoder network (Liu et al., 2024). Under regularity assumptions on the input and output functions and Lipschitz continuity of the operator, existing bounds yield only slow log-power scaling of the approximation and generalization errors with respect to model size and training-data size, respectively (Kovachki et al., 2024; Liu et al., 2026). This slow log-power scaling arises from the difficulty of learning in infinite-dimensional spaces, and it is shown to be optimal for Lipschitz functionals (Lanthaler & Stuart, 2026, Proposition 2.21). This slow log-power scaling can be improved to the power scaling law when the learning problem exhibits a low-dimensional structure (Liu et al., 2024; 2025; 2026).

For transformer-based operator learning, existing works have investigated the approximation properties through universal approximation theory (Calvello et al., 2025; Shih et al., 2025; Zappala & Bagherian, 2026; Furuya et al., 2026). However, these works do not provide a quantitative scaling between the approximation error and the network size, and the generalization error has not been analyzed in the literature.

To fill this gap, we develop a theoretical framework to answer the following question:

How does transformer-based operator learning error scale with model size and data size?

In this paper, we answer this question by establishing quantitative approximation and generalization theories of transformer-based operator learning. We impose mild Hölder continuity assumptions on the input and output functions, as well as on the target operator. The complexity of the input function class is characterized by the L^2 covering number, which aligns with classical complexity definitions on function/metric spaces and statistical learning theory (Hurewicz & Wallman, 2015; Kolmogorov & Tikhomirov, 1959; Wainwright, 2019). This complexity characterization arises naturally in scientific applications where the input functions are governed by a small number of physical parameters, such as parameterized initial or boundary conditions, source locations and amplitudes, or geometric and material parameters.

To the best of our knowledge, this paper is the first to theoretically characterize neural scaling laws for Transformer-based operator learning. Our main contributions are summarized as follows.

- **Approximation and model scaling.** On approximation theory, we explicitly construct a two-block softmax Transformer network to approximate γ-Hölder continuous operators in Theorem 1. When the transformer has model size N_{total}, the operator approximation error scales like $N_{\text{total}}^{-1/s}$ where the parameter s is specified in (17) with explicit dependence on the operator regularity, function regularity, and input function-space complexity.
- **Generalization and data scaling.** On generalization theory, we derive a generalization error bound for the empirical risk minimizer in Theorem 2. When a transformer with a proper architecture is trained on n input-output function pairs, the squared operator generalization error scales like $n^{-\frac{2}{2+s}}$, which characterizes a data scaling law between the operator prediction error and the number of training data size.

Our approximation and generalization results quantify the effects of operator regularity, function regularity, input-function space complexity, input-function discretization, model size, training-data size, and output observation noise. By leveraging the low-dimensional complexity of the input function space, we prove power scaling laws. This contrasts with existing quantitative analyses for feedforward-neural-network-based operator learning, and more broadly infinite-dimensional learning, which exhibit logarithmic-type power scaling laws when the low-dimensional input function complexity is not considered (Mhaskar & Hahm, 1997; Kovachki et al., 2024; Lanthaler & Stuart, 2026; Shi et al., 2025; Liu et al., 2026).

Our proof builds on a novel local-to-global approximation scheme based on two-level Softmax Partitions Of Unity (POU): one POU is over the input function space and the other POU is over the

output domain. The proof is then followed by an explicit realization of this two-level approximation by a two-block Transformer. This construction is naturally aligned with the Softmax attention mechanism and yields discretization-invariant outputs. The point-wise feedforward layers are only used for exact linear feature assembly.

Organization. The remainder of the paper is organized as follows. Section 2 introduces the operator-learning problem, data-generation setting, and Transformer architecture. Section 3 presents the approximation and generalization results, together with the resulting model and data scaling laws. Section 4 provides proof sketches for the main theorems. Section 5 presents numerical experiments on the Burgers' and KdV solution operators. Additional definitions, architectural details, extensions, complete proofs, supporting lemmas, and experimental details are in the appendices.

Notation. For a positive integer m, we write $[m] := \{1, \dots, m\}$. We use lower-case letters for scalars, bold lower-case letters for vectors, and bold upper-case letters for matrices. For a probability measure ρ, we write $\|f\|^2_{L^2(\rho)} := \int |f|^2 \, d\rho$. For a collection of network parameters Θ, $\|\Theta\|_\infty$ denotes the maximum absolute value of its entries, and $\mathrm{vec}(\mathbf{Z})$ denotes the column-wise vectorization of $\mathbf{Z}$.

2 Operator Learning Problem and Transformer Architecture

2.1 Problem setup

We consider supervised learning of a nonlinear operator $G : \mathcal{U} \to \mathcal{V}$ from finitely many evaluations of input functions and pointwise output queries. Let $\Omega_{\mathcal{U}} = [0,1]^{d_1}$ be the input domain and let $\Omega_{\mathcal{V}} \subseteq [0,1]^{d_2}$ be the output query domain. Here $\mathcal{U}$ is a class of scalar-valued functions on $\Omega_{\mathcal{U}}$, and $\mathcal{V}$ is a class of scalar-valued functions on $\Omega_{\mathcal{V}}$.

Setting 1 (Data Generation)**.** *A training data set $\mathcal{S}$ is generated under the following protocol:*

1. ***Function sampling.*** *The input functions $u_1, \dots, u_n$ are drawn i.i.d. from a probability distribution ρ_u supported on $\mathcal{U}$.*
2. ***Input discretization.*** *Let $X_{n_x} := (\mathbf{x}_1, \dots, \mathbf{x}_{n_x}) \subset \Omega_{\mathcal{U}}$ be an input grid, shared across all sampled functions and independent of the sampled functions. For each u_i, define its discretized representation by*
$$S_{n_x}(u_i) := \big((\mathbf{x}_1, u_i(\mathbf{x}_1)), \dots (\mathbf{x}_{n_x}, u_i(\mathbf{x}_{n_x}))\big). \tag{1}$$
3. ***Output query sampling.*** *The query points $\{\mathbf{y}_{i,j}\}_{j=1}^{n_y}$ are drawn i.i.d. from a probability distribution ρ_y supported on $\Omega_{\mathcal{V}}$, independently of $\{u_i\}_{i=1}^n$. The observed scalar outputs are*
$$v_{i,j} = G(u_i)(\mathbf{y}_{i,j}) + \xi_{i,j}, \qquad i \in [n],\ j \in [n_y],$$
where the noise variables $\{\xi_{i,j}\}_{i\in[n],\, j\in[n_y]}$ are independent, independent of all input functions and query points, and mean-zero sub-Gaussian with variance proxy σ^2.

The training sample can be written as $\mathcal{S} = \{(S_{n_x}(u_i), \mathbf{y}_{i,j}, v_{i,j}) : i \in [n],\ j \in [n_y]\}$. The data-generation protocol is illustrated in Figure 1.

For any predictor T mapping $(S_{n_x}(u), \mathbf{y})$ to a scalar, define the population risk by
$$\mathcal{L}(T) := \mathbb{E}_{u\sim\rho_u} \mathbb{E}_{\mathbf{y}\sim\rho_y} \left[\left(T(S_{n_x}(u), \mathbf{y}) - G(u)(\mathbf{y}) \right)^2 \right]. \tag{2}$$
For a hypothesis class $\mathcal{H}$, define the empirical risk on the training sample $\mathcal{S}$ by
$$\widehat{\mathcal{L}}_{\mathcal{S}}(T) := \tfrac{1}{n n_y} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n_y} \left(T(S_{n_x}(u_i), \mathbf{y}_{i,j}) - v_{i,j} \right)^2. \tag{3}$$
Let
$$\widehat{T} \in \operatorname*{arg\,min}_{T\in\mathcal{H}} \widehat{\mathcal{L}}_{\mathcal{S}}(T) \tag{4}$$
be an empirical risk minimizer. Since $\widehat{T}$ depends on the random training sample $\mathcal{S}$, our statistical objective is to bound the expected population risk
$$\mathbb{E}_{\mathcal{S}} \mathcal{L}(\widehat{T}) = \mathbb{E}_{\mathcal{S}} \mathbb{E}_{u\sim\rho_u} \mathbb{E}_{\mathbf{y}\sim\rho_y} \left[\left(\widehat{T}(S_{n_x}(u), \mathbf{y}) - G(u)(\mathbf{y}) \right)^2 \right], \tag{5}$$
where $\mathcal{H}$ is chosen as the Transformer hypothesis class introduced in Subsection 2.2.

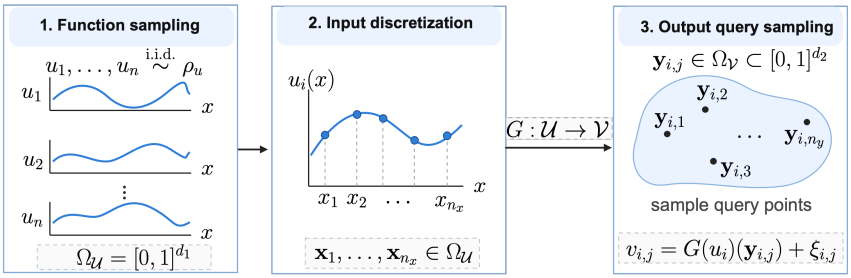

Figure 1: Illustration of the operator-learning data-generation setting. Each input function is observed on a shared input grid, while output values are queried at independently sampled locations.

2.2 Transformer Architecture

Given the discretized input function $S_{n_x}(u)$ defined in (1) and an output query $\mathbf{y} \in \Omega_{\mathcal{V}}$, we organize the input observations and the query into a sequence of tokens.

$$\mathbf{X}(u, \mathbf{y}) := \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{n_x} & \mathbf{0}_{d_1} & \cdots & \mathbf{0}_{d_1} \\ \mathbf{0}_{d_2} & \cdots & \mathbf{0}_{d_2} & \mathbf{y} & \cdots & \mathbf{0}_{d_2} \\ u(\mathbf{x}_1) & \cdots & u(\mathbf{x}_{n_x}) & 0 & \cdots & 0 \end{bmatrix}. \tag{6}$$

The first n_x columns encode the spatial locations and corresponding function values in $S_{n_x}(u)$, while one additional column encodes the output query $\mathbf{y}$.

A preprocessing map converts (6) into the initial hidden representation

$$\mathbf{Z}_0 = \mathcal{P}\big(S_{n_x}(u), \mathbf{y}\big) = \mathbf{W}_E \mathbf{X}(u, \mathbf{y}) + \mathbf{b}_E \mathbf{1}_P^\top + \mathbf{P}_{\text{pos}} \in \mathbb{R}^{D \times P}, \tag{7}$$

where $\mathbf{W}_E$ and $\mathbf{b}_E$ define a token-wise affine embedding and $\mathbf{P}_{\text{pos}}$ is a fixed structural-positional encoding. The explicit choice of the preprocessing map used in the constructive approximation proof is given in Appendix B.2.

The representation is then propagated through L encoder blocks. For $\ell \in [L]$, let A_ℓ denote a multi-head self-attention layer and F_ℓ a point-wise ReLU feedforward layer, and write

$$\widehat{\mathbf{Z}}_\ell = A_\ell(\mathbf{Z}_{\ell-1}), \qquad \mathbf{Z}_\ell = F_\ell(\widehat{\mathbf{Z}}_\ell). \tag{8}$$

The precise definitions of A_ℓ and F_ℓ are provided in Appendix A.2. The scalar prediction is obtained through the linear readout

$$T_\Theta\big(S_{n_x}(u), \mathbf{y}\big) = \mathbf{c}_{L+1}^\top \operatorname{vec}(\mathbf{Z}_L). \tag{9}$$

We next define the Transformer hypothesis class used in our analysis.

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 $\{H^\ell\}_{\ell=1}^L$, $\{d_k^\ell\}_{\ell=1}^L$, $\{d_v^\ell\}_{\ell=1}^L$, $\{d_{\text{ff}}^\ell\}_{\ell=1}^L \subset \mathbb{N}$, and parameter magnitude bound $M_{\max} > 0$, define*

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

$$= \left\{ T_\Theta \;\middle|\; \begin{array}{l} T_\Theta \text{ is an } L\text{-block encoder-only Transformer of the form (7)–(9),} \\ \text{with embedding dimension } D, \text{ sequence length } P, \text{ and parameter bound } \|\Theta\|_\infty \le M_{\max}, \\ \text{and for each block } \ell \in [L], \text{ there are } H^\ell \text{ heads, query/key dimension } d_k^\ell, \\ \text{value dimension } d_v^\ell, \text{ and FFN hidden width } d_{\text{ff}}^\ell \end{array} \right\}.$$

3 Main Results

We now present the main theoretical results. We first state the assumptions used throughout the analysis. Under these assumptions, we establish a quantitative approximation result for Transformer-based operator learning and derive the resulting model scaling law. We then establish a generalization bound for the empirical risk minimizer and derive the corresponding data scaling law.

3.1 ASSUMPTIONS

To establish the approximation and generalization theory, we introduce the following assumptions on the operator, the input function class, and the output function class.

Assumption 1 (Operator Continuity)**.** *The operator $G : \mathcal{U} \to \mathcal{V}$ satisfies a γ-Hölder continuity condition in the $L^2(\Omega_\mathcal{U})$ and $L^2(\rho_y)$ metrics. Namely, there exist constants $L_G > 0$ and $\gamma \in (0, 1]$ such that, for all $u_1, u_2 \in \mathcal{U}$,*

$$\|G(u_1) - G(u_2)\|_{L^2(\rho_y)} \le L_G \|u_1 - u_2\|^\gamma_{L^2(\Omega_\mathcal{U})}. \tag{10}$$

Assumption 2 (Input Function Class)**.** *The input function class $\mathcal{U}$ satisfies the following conditions.*

1. ***Smoothness.*** *There exist constants $\alpha > 0$ and $B_\mathcal{U} > 1$ such that*

$$\sup_{u\in\mathcal{U}} \|u\|_{C^\alpha(\Omega_\mathcal{U})} \le B_\mathcal{U}, \tag{11}$$

   where the Hölder norm is defined in Definition 2 of Appendix A.1.

2. ***Low-dimensional structure.*** *For every $r_u \in (0, 1]$, there exists a r_u-cover $\{u_k\}_{k=1}^{C_U} \subset \mathcal{U}$ under the $L^2(\Omega_\mathcal{U})$ norm, that is, $\sup_{u\in\mathcal{U}} \min_{k\in[C_U]} \|u - u_k\|_{L^2(\Omega_\mathcal{U})} \le r_u$. Moreover, the covering number satisfies*

$$C_U \le C_1 r_u^{-d_\mathcal{U}}, \tag{12}$$

   for some constants $C_1 \ge 1$ and $d_\mathcal{U} \ge 0$.

Assumption 3 (Output Function Class)**.** *There exist constants $\beta \in (0, 1]$ and $B_\mathcal{V} > 0$ such that*

$$\sup_{v\in\mathcal{V}} \|v\|_{C^\beta(\Omega_\mathcal{V})} \le B_\mathcal{V}. \tag{13}$$

Remark 1. *The assumptions above separate the three sources of complexity in the analysis. Assumption 1 is a stability condition for the operator: nearby input functions in $L^2(\Omega_\mathcal{U})$ have nearby outputs in $L^2(\rho_y)$. The C^α and C^β regularity assumptions for input and output functions in Assumption 2-1 and Assumption 3 are standard in operator learning theory (Lanthaler et al., 2022; Lanthaler & Stuart, 2026; Kovachki et al., 2024; Liu et al., 2024; 2026).*

The covering condition in Assumption 2-2 captures the low-dimensional structure of $\mathcal{U}$ under the $L^2(\Omega_\mathcal{U})$ metric. By introducing Assumption 2-2, we can incorporate the function complexity into the theoretical framework. Roughly speaking, $d_\mathcal{U}$ denotes the number of parameters in the input function class. Such intrinsic low-dimensional structure is not merely a theoretical assumption, but arises naturally in many scientific applications where input functions are determined by a small number of physical parameters, such as parametrized initial or boundary conditions, source locations and amplitudes, material coefficients, geometric parameters, or low-dimensional latent representations of random fields.

3.2 APPROXIMATION THEORY AND MODEL SCALING LAW

We first establish a quantitative approximation bound for the Transformer class in Definition 1.

Theorem 1 (Transformer Approximation for Operator Learning)**.** *In Setting 1, suppose Assumptions 1, 2, 3 hold and $d_\mathcal{U}/\gamma + d_2/\beta \ge d_1/\alpha$. For the input discretization, let X_{n_x} be a sampling grid constructed from tensor-product Gauss–Legendre points on a uniform partition of $\Omega_\mathcal{U} = [0, 1]^{d_1}$. Let $\epsilon_0 \in (0, e^{-1}]$ be a structural constant. For any $\epsilon \in (0, \epsilon_0)$, suppose the sequence length $P = C_U C_V + 1$ satisfies the compatibility condition $P \ge n_x + 2$. Then there exists a Transformer network*

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

such that

$$\sup_{u\in\mathcal{U}} \|T_\Theta(S_{n_x}(u), \cdot) - G(u)\|_{L^2(\rho_y)} \le \epsilon. \tag{14}$$

The network can be chosen with the following structural parameters:

- *To control the input-discretization error, choose $n_x = \Big\lceil (16 B_\mathcal{V} M_u C_{\text{quad}}/\epsilon)^{d_1/\alpha} \Big\rceil$.*
- *$L = 2$, $D = d_1 + d_2 + n_x + 9$, and the sequence length satisfies $P \le C_P \epsilon^{-d_\mathcal{U}/\gamma - d_2/\beta}$.*

- *In block $\ell = 1$, $H^1 = (n_x + 3)P + 2$, $d_k^1 = 5$, $d_v^1 = 2$, and $d_{\mathrm{ff}}^1 = 2D$.*
- *In block $\ell = 2$, $H^2 = 1$, $d_k^2 = d_v^2 = 1$, and $d_{\mathrm{ff}}^2 = 2D$.*

Here M_u is defined in (32) *of Appendix B.1, C_P is the positive constant defined in* (54) *of Appendix B.3, and C_{quad} is the quadrature constant in Lemma 3 of Appendix B.1. The maximum parameter magnitude $M_{\max} := \|\Theta\|_\infty$ and the total number of network parameters N_{total} satisfy*

$$M_{\max} \le C_{\mathrm{mag}} \epsilon^{-q_M} \left(\log \epsilon^{-1}\right)^{1+d_1/\alpha}, \tag{15}$$

$$N_{\mathrm{total}} \le C_N \epsilon^{-s} \left(\log \epsilon^{-1}\right)^{2d_1/\alpha}, \tag{16}$$

where

$$s := \frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta} + \frac{2d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) \tag{17}$$

and

$$q_M := \max\left\{2\left(\frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}\right), \frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + \frac{2}{\gamma}, \frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + \frac{2}{\beta}\right\}.$$

The positive constants C_{mag} and C_N are defined in (62) *and* (58) *in Appendix B.3.*

The proof of Theorem 1 is given in Appendix B.3. The theorem has the following implications.

- **Model scaling law.** Theorem 1 quantifies how the approximation accuracy improves as the Transformer model size increases. A model of size N_{total} can achieve an approximation error in the order of $N_{\mathrm{total}}^{-1/s}$ up to logarithmic factors. The scaling exponent $-1/s$ explicitly reflects the effect of the intrinsic input dimension $d_{\mathcal{U}}$, and the function domain dimensions d_1, d_2, and the function and operator regularity. By leveraging the input intrinsic dimension $d_{\mathcal{U}}$, Theorem 1 yields a power model scaling law. In contrast, without the exploration of the input intrinsic dimension, the model scaling law is proved in the order of $(\log N_{\mathrm{total}})^{-\mathrm{power}}$ for feedforward-type operator learning with (Kovachki et al., 2024; Liu et al., 2026; Lanthaler & Stuart, 2026). For band-limited input functions, the rate of convergence can be further improved in Appendix A.3.
- **First quantitative approximation result for transformer operator network.** Existing approximation theories for transformer-based operator learning establish universal approximation properties (Calvello et al., 2025; Shih et al., 2025; Zappala & Bagherian, 2026; Furuya et al., 2026), but do not explicitly quantify the relation between approximation accuracy and Transformer model size. To the best of our knowledge, Theorem 1 is the first quantitative approximation result for transformer-based operator learning that explicitly characterizes the scaling between the approximation error and the Transformer model size.

3.3 Generalization theory and data scaling law

We next turn from approximation to statistical learning. Our goal is to bound the expected population risk $\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T})$ in (5). To uniformly control the prediction error relative to the bounded target operator, we use a clipped Transformer class.

For the output bound $B_{\mathcal{V}} > 0$ in Assumption 3, define the clipping map $\Pi_{B_{\mathcal{V}}}(t) = \min\{\max\{t, -B_{\mathcal{V}}\}, B_{\mathcal{V}}\}$. For a fixed approximation tolerance ϵ, let $\mathcal{T}_\epsilon$ denote the Transformer class specified in Theorem 1, and define the corresponding clipped hypothesis class by $\mathcal{H}_\epsilon^{\mathrm{clip}} := \{\Pi_{B_{\mathcal{V}}} \circ T : T \in \mathcal{T}_\epsilon\}$. We then define the clipped empirical risk minimizer (ERM) by

$$\widehat{T}_\epsilon \in \operatorname*{arg\,min}_{T \in \mathcal{H}_\epsilon^{\mathrm{clip}}} \widehat{\mathcal{L}}_{\mathcal{S}}(T). \tag{18}$$

Theorem 2 (Generalization Error of the Clipped Transformer ERM)**.** *In Setting 1, suppose Assumptions 1, 2, 3 hold and $d_{\mathcal{U}}/\gamma + d_2/\beta \ge d_1/\alpha$. Let $n, n_y \ge 2$, and use the input discretization X_{n_x} in Theorem 1. For any $\epsilon \in (0, \epsilon_0)$, suppose the sequence-length compatibility condition $P = C_U C_V + 1 \ge n_x + 2$ is satisfied. Then the clipped ERM $\widehat{T}_\epsilon$ in* (18) *satisfies*

$$\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T}_\epsilon) \le C_{\mathrm{gen}} \left[\epsilon^2 + \left(1 + \frac{\sigma^2}{n_y}\right) \frac{\epsilon^{-s}\left(\log \frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1}}{n}\right], \tag{19}$$

where s is defined in (17)*, and the constant $C_{\mathrm{gen}} > 0$ is defined in* (64) *of Appendix C.2, which depends only on the structural quantities in the assumptions and σ, but not on ϵ, n, or n_y.*

Moreover, choose

$$\epsilon = \left[\frac{1 + \sigma^2/n_y}{n} (\log n)^{\frac{2d_1}{\alpha}+1} \right]^{\frac{1}{2+s}}.$$

Provided that this choice lies in $(0, \epsilon_0)$ and satisfies the preceding sequence-length compatibility condition, we have

$$\mathbb{E}_{\mathcal{S}} \mathcal{L}(\widehat{T}_\epsilon) \leq C_{\mathrm{rate}} \left[\frac{1 + \sigma^2/n_y}{n} (\log n)^{\frac{2d_1}{\alpha}+1} \right]^{\frac{2}{2+s}}, \tag{20}$$

where the constant $C_{\mathrm{rate}} := 2C_{\mathrm{gen}}$.

The proof of Theorem 2 is given in Appendix C.2. The theorem has the following implications.

- **Data scaling law.** Theorem 2 quantifies a power scaling law between the generalization/prediction error and the training data size n in (20). The mean squared generalization error is in the order of $n^{-\frac{2}{2+s}}$ up to logarithmic factors. The scaling exponent $-2/(2+s)$ characterizes the effect from the input intrinsic dimension $d_{\mathcal{U}}$, and the function domain dimensions d_1, d_2, and the function/operator regularity. The output query size n_y helps to reduce the noise contribution. A large n_y allows one to reduce the noise effect, since the upper bound in (20) depends on σ^2/n_y.
- **First quantitative generalization result for transformer operator network.** Empirical experiments on transformer-based operator learning have demonstrated systematic performance improvements as the training data size increases (Li et al., 2022; Hao et al., 2023), but a theoretical characterization of this data dependence remains underdeveloped. To the best of our knowledge, Theorem 2 provides the first quantitative generalization result that explicitly characterizes the data scaling law for transformer-based operator learning.

4 Proof Sketch

4.1 Proof sketch of Theorem 1.

The proof consists of two main steps, corresponding to Appendices B.1 and B.2. First, building on the two-level Softmax partition-of-unity (POU) framework for in-context learning introduced by Shi et al. (2026), we construct an oracle approximation of the target operator: one Softmax POU localizes the input function over the input function space, while the other localizes the output query over the output domain. Second, we show that this oracle approximation can be explicitly realized by a two-block Transformer, where the attention mechanism implements the localization and aggregation induced by the two-level POU constructions.

• **Step 1: Oracle approximation via two-level Softmax POU (Appendix B.1).** We first construct an oracle approximation of the target operator independently of the Transformer architecture. The construction is illustrated in Figure 2. Let $\{u_k\}_{k=1}^{C_U} \subset \mathcal{U}$ be an r_u-cover of the input function class and let $\{\mathbf{z}_l\}_{l=1}^{C_V} \subset \Omega_{\mathcal{V}}$ be an r_y-cover of the output domain. We use the localization parameters M_u and M_y defined in (32) and (36) of Appendix B.1. The input-space Softmax POU is constructed from the discretized inner products

$$\widetilde{\beta}_{k,n_x}(u) := \frac{\exp\left(2M_u \langle u, u_k \rangle_{n_x} - M_u \|u_k\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}{\sum_{k'=1}^{C_U} \exp\left(2M_u \langle u, u_{k'} \rangle_{n_x} - M_u \|u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}, \tag{21}$$

while the output-domain Softmax POU is

$$\widetilde{\eta}_l(\mathbf{y}) := \frac{\exp\left(2M_y \mathbf{y}^\top \mathbf{z}_l - M_y \|\mathbf{z}_l\|_2^2\right)}{\sum_{l'=1}^{C_V} \exp\left(2M_y \mathbf{y}^\top \mathbf{z}_{l'} - M_y \|\mathbf{z}_{l'}\|_2^2\right)}. \tag{22}$$

Combining the two partitions gives the oracle approximation

$$\widetilde{G}_{r_u, r_y, n_x}(u)(\mathbf{y}) = \sum_{k=1}^{C_U} \sum_{l=1}^{C_V} G(u_k)(\mathbf{z}_l) \widetilde{\beta}_{k,n_x}(u) \widetilde{\eta}_l(\mathbf{y}). \tag{23}$$

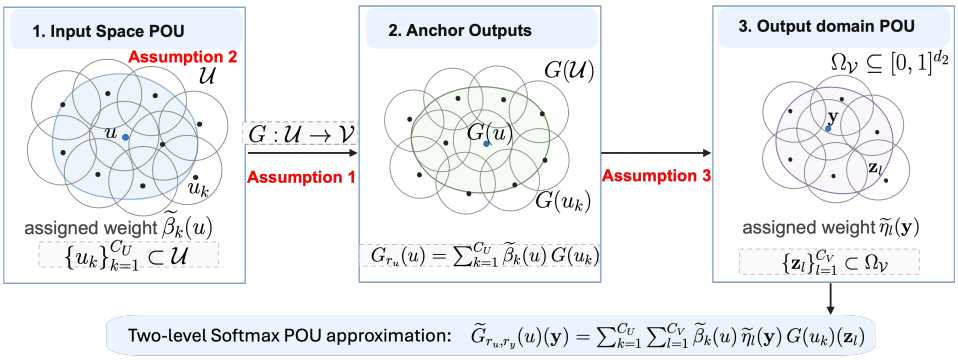

Figure 2: Illustration of the two-level Softmax POU approximation. The input-space POU localizes u over function-space anchors $\{u_k\}_{k=1}^{C_U}$, while the output-domain POU localizes the query $\mathbf{y}$ over anchors $\{\mathbf{z}_l\}_{l=1}^{C_V}$. Their combination yields the oracle approximation in (23).

Error control for the oracle construction (Lemmas 1–5 in Appendix B.1). The supporting lemmas quantify the input-space localization, output-domain localization, and input-discretization errors. Together, they show that the oracle approximation converges to the target operator as r_u, r_y, and the input discretization are refined.

• **Step 2: Transformer realization of the oracle approximation (Appendix B.2).** We next show that the oracle approximation (23) can be realized by a two-block Transformer in Section 2.2.

Pair-token preprocessing (Lemma 6 in Appendix B.2). The preprocessing map in (7) is chosen so that the first $P - 1 = C_U C_V$ active tokens are indexed by the input-output anchor pairs $(u_k, \mathbf{z}_l) \in \{u_k\}_{k=1}^{C_U} \times \{\mathbf{z}_l\}_{l=1}^{C_V}$.

Linear feature extraction and joint-logit assembly (Lemmas 7 and 8 in Appendix B.2). The first attention layer A_1 computes in parallel the linear features appearing in the input- and output-localization logits in (21) and (22), together with the corresponding anchor values $G(u_k)(\mathbf{z}_l)$. The point-wise feedforward layer F_1 assembles these features into the joint localization logit $2M_u\langle u, u_k\rangle_{n_x} - M_u\|u_k\|_{L^2}^2 + 2M_y\mathbf{y}^\top\mathbf{z}_l - M_y\|\mathbf{z}_l\|_2^2$ for each anchor pair (k, l).

Softmax aggregation (Lemma 9 in Appendix B.2). The second attention layer A_2 applies Softmax to the joint localization logits. Since each joint logit is the sum of the input-space and output-domain logits, the resulting normalized weight factorizes as $\widetilde{\beta}_{k,n_x}(u)\widetilde{\eta}_l(\mathbf{y})$. Using $G(u_k)(\mathbf{z}_l)$ as the corresponding values, A_2 therefore performs exactly the two-level aggregation in (23). The second feedforward layer F_2 is chosen as the identity, and the linear readout in (9) extracts the scalar output.

The supporting lemmas in Appendix B.2 show that this two-block construction realizes the oracle approximation to the accuracy required in Theorem 1. The full proof in Appendix B.3 combines this realization with the oracle approximation from Step 1 to yield the approximation bound (14) and, by tracking the resulting architecture size, the model scaling law (16).

4.2 Proof sketch of Theorem 2.

The proof consists of two main steps, corresponding to Appendices C.1 and C.2. First, we control the variance term, which measures the statistical estimation error of the empirical risk minimizer, through a covering-number bound for the clipped Transformer hypothesis class. Second, the approximation result in Theorem 1 controls the bias term by showing that the Transformer class can approximate the target operator with squared error of order ϵ^2. The proof of Theorem 2 combines these two bounds and balances them to obtain the resulting data scaling law.

Covering-number control of the variance (Lemma 10 in Appendix C.1). Using the covering number defined in Definition 3 of Appendix A.1, Lemma 10 gives

$$\log \mathcal{N}\left(\eta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right) \le N_{\mathrm{total}} \log\left(503552 P^7 D^{17} M_{\max}^{21} B_{\mathcal{U}}^5/\eta\right).$$

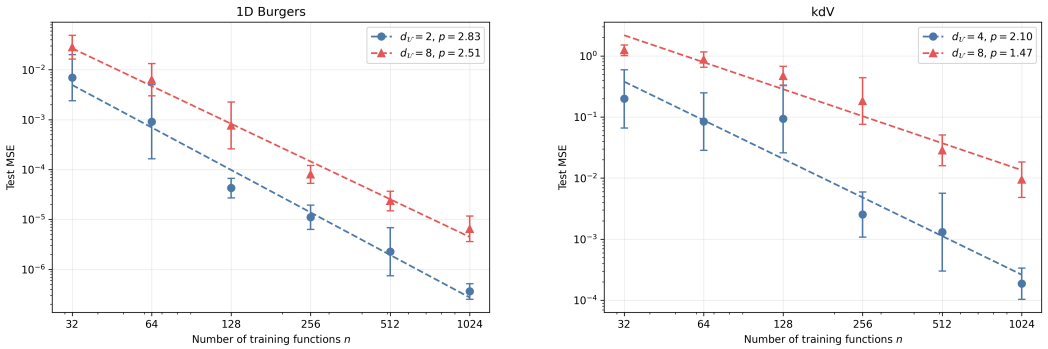

Figure 3: Empirical data scaling for Burgers' (left) and KdV (right). Markers show the geometric mean test MSE over five repeats, with error bars corresponding to one sample standard deviation of the log MSE; dashed lines show the fitted power law $\mathrm{MSE} \approx Cn^{-p}$.

Bias–variance tradeoff (Appendix C.2). The approximation bound (14) controls the bias term, which is of order ϵ^2. The covering-number estimate above controls the variance term through the empirical risk minimization property and the sub-Gaussian noise assumption. Combining these two gives (19). Balancing the two terms by the choice of ϵ in Theorem 2 yields the data scaling law (20).

5 Experiments

We use two nonlinear PDE solution operators to test a central implication of Theorem 2: whether increasing the intrinsic dimension $d_{\mathcal{U}}$ of the input function class is associated with slower empirical data scaling. We consider the one-dimensional viscous Burgers' equation $u_t + uu_x = \nu u_{xx}$ and the Korteweg–de Vries (KdV) equation $u_t + 6uu_x + u_{xxx} = 0$, representing dissipative and dispersive dynamics, respectively, and learn the final-time solution operator $G : u_0(\cdot) \mapsto u(\cdot, T)$.

For Burgers', we set $\nu = 0.02$ and $T = 0.05$ and consider parameterized compactly supported bump families with $d_{\mathcal{U}} \in \{2, 8\}$. For KdV, we use exact multi-soliton families on $x \in [0, 44]$ at $T = 1$ with $d_{\mathcal{U}} \in \{4, 8\}$.

For both equations, we use $n \in \{32, 64, 128, 256, 512, 1024\}$ independently sampled training functions, while keeping the Transformer architecture and training protocol fixed within each problem. Test MSE is evaluated in physical units over 128 held-out functions and 1024 uniformly spaced spatial points. Common implementation and evaluation details are provided in Appendix E.1, with PDE-specific input families and additional details in Appendices E.2 and E.3.

Figure 3 shows clear power-law decay over $n = 32$–1024 for both equations. Burgers' yields $p = 2.83$ for $d_{\mathcal{U}} = 2$ and $p = 2.51$ for $d_{\mathcal{U}} = 8$, while KdV yields $p = 2.10$ for $d_{\mathcal{U}} = 4$ and $p = 1.47$ for $d_{\mathcal{U}} = 8$. Despite the distinct dissipative and dispersive dynamics, both problems exhibit the same ordering: the higher-dimensional input family has the smaller empirical decay exponent. This trend is qualitatively consistent with Theorem 2, where increasing $d_{\mathcal{U}}$ increases s and moves the exponent $-2/(2+s)$ toward zero; the empirical exponents are not interpreted as quantitative estimates of the theoretical upper-bound rate.

6 Conclusion

Conclusion: We establish quantitative approximation and generalization theories for Transformer-based operator learning through a two-level Softmax POU construction and its explicit Transformer realization. The resulting power model and data scaling laws explicitly capture the operator regularity and the regularity and complexity of the input and output function spaces. Experiments on Burgers' and KdV equations exhibit the predicted quantitative data scaling law and its dependence on the input intrinsic dimension.

Limitation: The resulting scaling laws in this paper are constructive upper bounds and are not intended to characterize optimal exponents or the performance of every trained architecture. Sharpening these bounds and understanding how architectural choices and optimization affect the observed scaling behavior are future-work directions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing writing, finding related literature for references, designing the research and experiment idea, examining mathematical derivations, and implementation of numerical experiments. In particular, AI tools were used to assist with the design and analysis of the numerical experiments, the development and debugging of code for synthetic PDE datasets including Burgers' and KdV, and the verification and refinement of theoretical arguments. All AI-assisted mathematical arguments, code, numerical results, and manuscript content were independently reviewed and verified by the authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Complete proofs of the theoretical results are provided in the appendix. Additional details on the PDE benchmarks, input-function families, Transformer architecture, training protocol, and evaluation procedure are provided in Appendix E.

REFERENCES

Andrey Bryutkin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, and Angelica Aviles-Rivero. Hamlet: Graph transformer neural operator for partial differential equations. *arXiv preprint arXiv:2402.03541*, 2024.

Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *Journal of Machine Learning Research*, 26(300):1–52, 2025.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Junfeng Chen and Kailiang Wu. Positional knowledge is all you need: Position-induced transformer (pit) for operator learning. *arXiv preprint arXiv:2405.09285*, 2024.

Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, Orlando, FL, 2 edition, 1984. ISBN 9780122063602. doi: 10.1016/C2013-0-10566-1.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831. PMLR, 2022.

Takashi Furuya, David Mis, Ivan Dokmanić, Maarten V de Hoop, and Matti Lassas. Function graph transformers universally approximate operators between function spaces. *arXiv preprint arXiv:2605.17968*, 2026.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569, 2023.

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.

Witold Hurewicz and Henry Wallman. Dimension theory (pms-4). 2015.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Andrei Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

Nikola B Kovachki, Samuel Lanthaler, and Andrew M Stuart. Operator learning: Algorithms and analysis. *Handbook of Numerical Analysis*, 25:419–467, 2024.

Samuel Lanthaler and Andrew M Stuart. The parametric complexity of operator learning. *IMA Journal of Numerical Analysis*, 46(2):647–712, 2026.

Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and its Applications*, 6(1):tnac001, 2022.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023.

Hao Liu, Haizhao Yang, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Deep nonparametric estimation of operators between infinite dimensional spaces. *Journal of Machine Learning Research*, 25 (24):1–67, 2024.

Hao Liu, Biraj Dahal, Rongjie Lai, and Wenjing Liao. Generalization error guaranteed auto-encoder-based nonlinear model reduction for operator learning. *Applied and Computational Harmonic Analysis*, 74:101717, 2025.

Hao Liu, Zecheng Zhang, Wenjing Liao, and Hayden Schaeffer. Neural scaling laws of deep relu and deep operator network: A theoretical study, 2026. arXiv:2410.00357v2.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Xihaier Luo, Xiaoning Qian, and Byung-Jun Yoon. Hierarchical neural operator transformer with learnable frequency-aware loss prior for arbitrary-scale super-resolution. *arXiv preprint arXiv:2405.12202*, 2024.

Hrushikesh Narhar Mhaskar and Nahmwoo Hahm. Neural networks for functional approximation and system identification. *Neural Computation*, 9(1):143–159, 1997.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Zhongjie Shi and Wenjing Liao. Learning theory of transformers: Local-to-global approximation via softmax partition of unity. In *Advances in Neural Information Processing Systems*, 2026.

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.

Zhongjie Shi, Rongjie Lai, Alexander Cloninger, and Wenjing Liao. Transformers as cross-task learners: Shared structure drives sample efficiency in in-context learning. *arXiv preprint arXiv:2609.29060*, 2026.

Benjamin Shih, Ahmad Peyvan, Zhongqiang Zhang, and George Em Karniadakis. Transformers as neural operators for solutions of differential equations with finite regularity. *Computer Methods in Applied Mechanics and Engineering*, 434:117560, 2025.

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.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Yahong Yang, Zecheng Zhang, Wenjing Liao, and Hao Liu. Generalization guarantees for multi-input neural operator learning in sobolev spaces. *arXiv preprint arXiv:2606.17419*, 2026.

Emanuele Zappala and Maryam Bagherian. Universal approximation of operators with transformers and neural integral operators. *Axioms*, 15(9):676, 2026.

APPENDIX

A ADDITIONAL DEFINITIONS, TRANSFORMER ARCHITECTURE, AND EXTENSIONS

This section collects additional definitions and Transformer architecture details used throughout the analysis, and provides the extension of our approximation result to band-limited input functions.

A.1 ADDITIONAL DEFINITIONS

Definition 2 (Hölder Norm). *Let $\alpha = m_\alpha + \nu_\alpha$, where $m_\alpha \in \mathbb{N}_0$ and $\nu_\alpha \in (0, 1]$. For a function $f : \Omega \to \mathbb{R}$, define*

$$\|f\|_{C^\alpha(\Omega)} := \sum_{|\mathbf{r}| \le m_\alpha} \|\partial^{\mathbf{r}} f\|_{L^\infty(\Omega)} + \max_{|\mathbf{r}|=m_\alpha} \sup_{\mathbf{x} \neq \mathbf{x}'} \frac{|\partial^{\mathbf{r}} f(\mathbf{x}) - \partial^{\mathbf{r}} f(\mathbf{x}')|}{\|\mathbf{x} - \mathbf{x}'\|_2^{\nu_\alpha}}, \tag{24}$$

where $\mathbf{r}$ is a multi-index. The same convention is used for $C^\beta(\Omega_{\mathcal{V}})$.

Definition 3 (Covering Number). *Let $(\mathcal{F}, d)$ be a metric space and let $\eta > 0$. The covering number $\mathcal{N}(\eta, \mathcal{F}, d)$ is the minimum cardinality of an η-cover of $\mathcal{F}$ with respect to d.*

For scalar-valued predictors considered in this paper, we use the uniform norm

$$\|T\|_{\infty,\infty} := \sup_{u \in \mathcal{U},\, \mathbf{y} \in \Omega_{\mathcal{V}}} |T(S_{n_x}(u), \mathbf{y})|.$$

Definition 4 (Softmax Map). *For $\mathbf{a} = (a_1, \dots, a_m)^\top \in \mathbb{R}^m$, define*

$$[\mathrm{softmax}(\mathbf{a})]_j := \frac{\exp(a_j)}{\sum_{r=1}^m \exp(a_r)}, \qquad j \in [m]. \tag{25}$$

A.2 TRANSFORMER ARCHITECTURE

We provide the precise definitions of the multi-head self-attention and point-wise feedforward maps used in Section 2.2. For $\mathbf{Z} = [\mathbf{z}^1, \dots, \mathbf{z}^P] \in \mathbb{R}^{D \times P}$, the h-th attention head in block ℓ uses

$$\mathbf{Q}_\ell^h, \mathbf{K}_\ell^h \in \mathbb{R}^{d_k^\ell \times D}, \qquad \mathbf{V}_\ell^h \in \mathbb{R}^{d_v^\ell \times D}.$$

The attention score matrix is defined by

$$\mathbf{S}_\ell^h(\mathbf{Z}) := (\mathbf{K}_\ell^h \mathbf{Z})^\top (\mathbf{Q}_\ell^h \mathbf{Z}) \in \mathbb{R}^{P \times P}. \tag{26}$$

Using the Softmax map defined in (25), the corresponding column-wise attention matrix is

$$[\mathbf{A}_\ell^h(\mathbf{Z})]_{:,j} := \mathrm{softmax}\left([\mathbf{S}_\ell^h(\mathbf{Z})]_{:,j}\right), \qquad j \in [P]. \tag{27}$$

The h-th attention head is then

$$\mathrm{head}_\ell^h(\mathbf{Z}) := \mathbf{V}_\ell^h \mathbf{Z}\, \mathbf{A}_\ell^h(\mathbf{Z}) \in \mathbb{R}^{d_v^\ell \times P}. \tag{28}$$

With $\mathbf{W}_\ell^O \in \mathbb{R}^{D \times (H^\ell d_v^\ell)}$, the multi-head self-attention map is

$$A_\ell(\mathbf{Z}) := \mathbf{W}_\ell^O \begin{bmatrix} \mathrm{head}_\ell^1(\mathbf{Z}) \\ \vdots \\ \mathrm{head}_\ell^{H^\ell}(\mathbf{Z}) \end{bmatrix} \in \mathbb{R}^{D \times P}. \tag{29}$$

The point-wise feedforward map is defined by

$$F_\ell(\mathbf{Z}) := \mathbf{W}_\ell^2 \sigma\big(\mathbf{W}_\ell^1 \mathbf{Z} + \mathbf{b}_\ell^1 \mathbf{1}_P^\top\big) + \mathbf{b}_\ell^2 \mathbf{1}_P^\top, \tag{30}$$

where

$$\mathbf{W}_\ell^1 \in \mathbb{R}^{d_{\mathrm{ff}}^\ell \times D}, \qquad \mathbf{W}_\ell^2 \in \mathbb{R}^{D \times d_{\mathrm{ff}}^\ell},$$

and $\sigma(t) = \max\{t, 0\}$ is applied entrywise.

Therefore, the ℓ-th encoder block used in our analysis is

$$\widehat{\mathbf{Z}}_\ell = A_\ell(\mathbf{Z}_{\ell-1}), \qquad \mathbf{Z}_\ell = F_\ell(\widehat{\mathbf{Z}}_\ell). \tag{31}$$

A.3 Extension to Band-limited Input Functions

Recall from Theorem 1 that the complexity exponent s defined in (17) is

$$s = \frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta} + \frac{2d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right).$$

The last term, $\frac{2d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right)$, arises from the discretization of the input functions. Indeed, for a general C^α input class, controlling the quadrature error requires the number of input samples n_x to grow as the target accuracy ϵ decreases. Since the Transformer construction in Theorem 1 has parameter complexity of order $n_x^2 P$, this discretization requirement contributes directly to the model scaling exponent.

We next consider a band-limited input class, for which the required functional inner products can be recovered exactly from a fixed number of samples. In this case, n_x no longer grows with ϵ, and the input-discretization contribution disappears from the polynomial scaling exponent.

Corollary 1 (Scaling for Band-limited Input Functions)**.** *Suppose the assumptions of Theorem 1 hold. In addition, assume that every $u \in \mathcal{U}$ is periodic on $\Omega_{\mathcal{U}} = [0,1]^{d_1}$ and has Fourier support contained in $\left\{\mathbf{k} \in \mathbb{Z}^{d_1} : \|\mathbf{k}\|_\infty \le B\right\}$, where B is fixed independently of ϵ.*

Then the input grid in Setting 1 can be chosen with $n_x = n_x(B, d_1)$ independent of ϵ such that for $u, u_k \in \mathcal{U}$, $\langle u, u_k\rangle_{n_x} = \langle u, u_k\rangle_{L^2(\Omega_{\mathcal{U}})}$. So the quadrature term in Lemma 5 vanishes. The Transformer construction of Theorem 1 then satisfies $N_{\text{total}} \le C_{\text{band}}\epsilon^{-s_{\text{band}}}$, up to logarithmic factors, where $s_{\text{band}} := \frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}$.

Why the rate improves For $u, u_k \in \mathcal{U}$, the product uu_k has Fourier bandwidth at most $2B$. Hence a sufficiently fine tensor-product uniform grid integrates uu_k exactly, so the quadrature error in Lemma 3 vanishes and n_x can be chosen independently of ϵ. The remaining approximation argument is unchanged. In particular, $P = C_U C_V + 1 = O\left(\epsilon^{-\frac{d_{\mathcal{U}}}{\gamma} - \frac{d_2}{\beta}}\right)$. Since n_x is now fixed, the parameter bound $N_{\text{total}} = O(n_x^2 P)$ reduces to $N_{\text{total}} = O(P)$, which gives us s_{band}.

B Proof of Theorem 1: Approximation and Model Scaling

Following the proof sketch in Section 4, Appendix B.1 develops the two-level Softmax POU oracle approximation, Appendix B.2 establishes its Transformer realization, and Appendix B.3 combines these ingredients to prove Theorem 1 and derive the model scaling law.

B.1 Oracle Approximation via Two-Level Softmax POU

This subsection provides the supporting lemmas for Step 1 of the proof sketch in Section 4. Recall that the oracle approximation is constructed using an input-function-space Softmax POU and an output-domain Softmax POU, with localization parameters M_u and M_y defined in (32) and (36), respectively.

Lemma 1 (Input-function-space Softmax POU Approximation)**.** *Suppose Assumptions 1, 2, and 3 hold. Fix $r_u \in (0, 1]$. Let $\{u_k\}_{k=1}^{C_U} \subset \mathcal{U}$ be an r_u-cover of $\mathcal{U}$ in $L^2(\Omega_{\mathcal{U}})$, where $C_U \le C_1 r_u^{-d_{\mathcal{U}}}$. Assume r_u is sufficiently small so that $2B_{\mathcal{V}} C_U \ge L_G r_u^\gamma$, and define*

$$M_u := \frac{1}{3r_u^2}\log\left(\frac{2B_{\mathcal{V}} C_U}{L_G r_u^\gamma}\right). \tag{32}$$

For $k \in [C_U]$, define the continuous input-space Softmax POU weight by

$$\widetilde{\beta}_k(u) := \frac{\exp\left(2M_u\langle u, u_k\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_k\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}{\sum_{k'=1}^{C_U}\exp\left(2M_u\langle u, u_{k'}\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}, \tag{33}$$

where $\langle u, u_k \rangle_{L^2} = \int_{\Omega_\mathcal{U}} u(\mathbf{x}) u_k(\mathbf{x}) d\mathbf{x}$. *Define the input-space POU approximation by*

$$G_{r_u}(u) := \sum_{k=1}^{C_U} \widetilde{\beta}_k(u) G(u_k). \tag{34}$$

Then

$$\sup_{u \in \mathcal{U}} \|G(u) - G_{r_u}(u)\|_{L^2(\rho_y)} \leq (2^\gamma + 1) L_G r_u^\gamma. \tag{35}$$

The proof of Lemma 1 is given in Appendix D.1.1.

Since each expert $G(u_k)$ is β-Hölder continuous (Assumption 3), we next approximate it using an output-domain Softmax POU over $\Omega_\mathcal{V}$.

Lemma 2 (Output-domain POU Approximation)**.** *Suppose Assumption 3 holds. Fix* $r_y \in (0, 1]$. *Let* $\{\mathbf{z}_l\}_{l=1}^{C_V} \subset \Omega_\mathcal{V}$ *be an* r_y*-cover of* $\Omega_\mathcal{V}$ *under the Euclidean metric, and define*

$$M_y := \frac{1}{3r_y^2} \log\left(\frac{2C_V}{r_y^\beta}\right). \tag{36}$$

Let the output-domain Softmax POU weights $\{\widetilde{\eta}_l\}_{l=1}^{C_V}$ *be defined as in* (22)*.*

For each $k \in [C_U]$*, define the output-domain POU approximation by*

$$\widetilde{v}_k(\mathbf{y}) := \sum_{l=1}^{C_V} G(u_k)(\mathbf{z}_l) \widetilde{\eta}_l(\mathbf{y}). \tag{37}$$

Then

$$\sup_{k \in [C_U]} \|G(u_k) - \widetilde{v}_k\|_{L^\infty(\Omega_\mathcal{V})} \leq (2^\beta + 1) B_\mathcal{V} r_y^\beta. \tag{38}$$

Lemma 2 is proved in Appendix D.1.2.

Combining the input-space and output-domain POU approximations, we define the continuous two-level Softmax POU approximation by

$$\widetilde{G}_{r_u, r_y}(u)(\mathbf{y}) := \sum_{k=1}^{C_U} \sum_{l=1}^{C_V} \widetilde{\beta}_k(u) \widetilde{\eta}_l(\mathbf{y}) G(u_k)(\mathbf{z}_l). \tag{39}$$

The input-space weights in (33) depend on functional inner products. Since the Transformer observes u only through its values on a fixed input grid, we next replace these inner products by deterministic quadrature.

Lemma 3 (Quadrature Error for Input Anchors)**.** *Suppose Assumption 2 holds. Write* $\alpha = m_\alpha + \nu$ *for* $m_\alpha \in \mathbb{N}_0$*, and* $\nu \in (0, 1]$*. Let* $s_{\mathrm{quad}} := \left\lceil \frac{m_\alpha + 1}{2} \right\rceil$*. For every* $n_x \geq (2s_{\mathrm{quad}})^{d_1}$*, define* $q := \left\lfloor \frac{n_x^{1/d_1}}{s_{\mathrm{quad}}} \right\rfloor$*.*

Partition $\Omega_\mathcal{U} = [0, 1]^{d_1}$ *into* q^{d_1} *congruent cubes and apply the tensor-product* s_{quad}*-point Gauss–Legendre quadrature rule on each cube. This gives* $(s_{\mathrm{quad}} q)^{d_1} \leq n_x$ *quadrature nodes. If* $(s_{\mathrm{quad}} q)^{d_1} < n_x$*, append distinct points with zero weights so that the resulting fixed input grid has exactly* n_x *points,* X_{n_x} *in Setting 1. The associated quadrature weights* $w_1, \ldots, w_{n_x}$ *are non-negative and satisfy*

$$w_j \geq 0, \qquad \sum_{j=1}^{n_x} w_j = 1, \tag{40}$$

Moreover, for every $h \in C^\alpha(\Omega_\mathcal{U})$*,*

$$\left| \int_{\Omega_\mathcal{U}} h(\mathbf{x})\, d\mathbf{x} - \sum_{j=1}^{n_x} w_j h(\mathbf{x}_j) \right| \leq C_{\mathrm{quad},0} \|h\|_{C^\alpha(\Omega_\mathcal{U})} n_x^{-\alpha/d_1},$$

where $C_{\mathrm{quad},0} > 0$ depends only on d_1 and α. Consequently, for the input-space cover $\{u_k\}_{k=1}^{C_U} \subset \mathcal{U}$ used in Lemma 1,

$$\sup_{u\in\mathcal{U}} \max_{k\in[C_U]} \left|\langle u, u_k\rangle_{L^2(\Omega_{\mathcal{U}})} - \langle u, u_k\rangle_{n_x}\right| \le C_{\mathrm{quad}} n_x^{-\alpha/d_1}, \tag{41}$$

where $\langle u, u_k\rangle_{n_x} = \sum_{j=1}^{n_x} w_j u(\mathbf{x}_j) u_k(\mathbf{x}_j)$, and $C_{\mathrm{quad}} > 0$, defined explicitly in (67), depends only on d_1, α, $B_{\mathcal{U}}$, and the quadrature construction, but not on u, u_k, C_U, or n_x.

Lemma 3 is proved in Appendix D.1.3.

We use the following standard Lipschitz stability property (Edelman et al., 2022, Corollary A.7).

Lemma 4 (Softmax Lipschitz Stability)**.** *For any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$,*

$$\|\operatorname{softmax}(\mathbf{a}) - \operatorname{softmax}(\mathbf{b})\|_1 \le 2\|\mathbf{a} - \mathbf{b}\|_\infty.$$

Combining the error bounds in Lemmas 1, 2, and 3, we obtain the following error bound for the discrete oracle approximation (23).

Lemma 5 (Total Approximation Error)**.** *Suppose Assumptions 1, 2, and 3 hold, and suppose the fixed quadrature rule satisfies Lemma 3. Let $r_u, r_y \in (0, 1]$, with r_u satisfying the condition in Lemma 1, and let M_u and M_y be defined as in (32) and (36). Then*

$$\sup_{u\in\mathcal{U}} \left\| G(u) - \widetilde{G}_{r_u, r_y, n_x}(u) \right\|_{L^2(\rho_y)} \le (2^\gamma+1) L_G r_u^\gamma + (2^\beta+1) B_{\mathcal{V}} r_y^\beta + 4 B_{\mathcal{V}} M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}. \tag{42}$$

Lemma 5 is proved in Appendix D.1.4.

Remark 2. *In particular, since $M_u = \frac{1}{3r_u^2} \log\left(\frac{2B_{\mathcal{V}} C_U}{L_G r_u^\gamma}\right)$ and $C_U \le C_1 r_u^{-d_{\mathcal{U}}}$, we have $\log C_U \le \log C_1 + d_{\mathcal{U}} \log \frac{1}{r_u}$. Therefore, the dependence of the required input grid size n_x on the intrinsic dimension $d_{\mathcal{U}}$ arises through the logarithmic factor in M_u.*

B.2 Transformer Realization of the Oracle Approximation

We now construct a Transformer network with $L = 2$ encoder blocks to realize the discrete two-level Softmax POU approximation. Let the embedding dimension be $D = d_1 + d_2 + n_x + 9$, and let the total sequence length be $P = C_U C_V + 1$. We assume $P \ge n_x + 2$, so that the first n_x columns store the discretized input information, the $(n_x + 1)$-st column stores the query point, and the last column is reserved as the null column. After the first encoder block, the first $P - 1 = C_U C_V$ active columns are indexed by anchor pairs $(k, l) \in [C_U] \times [C_V]$, with $j(k, l) = (k - 1)C_V + l$.

Lemma 6 (Pair-token Preprocessing)**.** *Let $S_{n_x}(u)$ be the discretized representation of $u \in \mathcal{U}$, and let $\mathbf{y} \in \Omega_{\mathcal{V}}$ be the query point. Set*

$$\theta_j := \frac{2\pi j}{P}, \qquad j \in [P].$$

There exists a fixed preprocessing operator $\mathcal{P}$ such that $\mathbf{Z}_0 = \mathcal{P}(S_{n_x}(u), \mathbf{y}) \in \mathbb{R}^{D\times P}$ has the form

$$\mathbf{Z}_0 = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{n_x} & \mathbf{0}_{d_1} & \mathbf{0}_{d_1\times(P-n_x-2)} & \mathbf{0}_{d_1} \\ \mathbf{0}_{d_2} & \cdots & \mathbf{0}_{d_2} & \mathbf{y} & \mathbf{0}_{d_2\times(P-n_x-2)} & \mathbf{0}_{d_2} \\ u(\mathbf{x}_1) & \cdots & u(\mathbf{x}_{n_x}) & 0 & \mathbf{0}_{1\times(P-n_x-2)} & 0 \\ 0 & \cdots & 0 & 0 & \mathbf{0}_{1\times(P-n_x-2)} & 1 \\ \mathbf{0}_{n_x+3} & \cdots & \mathbf{0}_{n_x+3} & \mathbf{0}_{n_x+3} & \mathbf{0}_{(n_x+3)\times(P-n_x-2)} & \mathbf{0}_{n_x+3} \\ 1 & \cdots & 1 & 1 & \mathbf{1}_{1\times(P-n_x-2)} & 1 \\ 1 & \cdots & 1 & 1 & \mathbf{1}_{1\times(P-n_x-2)} & 0 \\ \sin(\theta_1) & \cdots & \sin(\theta_{n_x}) & \sin(\theta_{n_x+1}) & \sin(\theta_{n_x+2}), \ldots, \sin(\theta_{P-1}) & \sin(\theta_P) \\ \cos(\theta_1) & \cdots & \cos(\theta_{n_x}) & \cos(\theta_{n_x+1}) & \cos(\theta_{n_x+2}), \ldots, \cos(\theta_{P-1}) & \cos(\theta_P) \end{bmatrix}. \tag{43}$$

The $(d_1 + d_2 + 2)$-th row is the null-column indicator $(0, \ldots, 0, 1)$. This row is used only for routing in the first MHA block and is not preserved after that block.

Lemma 6 is proved in Appendix D.2.1.

Lemma 7 (Parallel Pair-token Feature Construction via MHA). *Let $\mathbf{Z}_0 \in \mathbb{R}^{D\times P}$ be defined as in Lemma 6. For every active column $j \in [P-1]$, write*

$$k_j := \left\lceil \frac{j}{C_V} \right\rceil, \qquad l_j := j - (k_j - 1)C_V.$$

Define the target feature terms

$$\begin{aligned} B_{i,k}(u) &:= 2M_u w_i u_k(\mathbf{x}_i)u(\mathbf{x}_i), && i \in [n_x],\ k \in [C_U], \\ N_k &:= M_u \|u_k\|^2_{L^2(\Omega_{\mathcal{U}})}, && k \in [C_U], \\ Q_l(\mathbf{y}) &:= 2M_y \mathbf{y}^\top \mathbf{z}_l - M_y \|\mathbf{z}_l\|_2^2, && l \in [C_V], \\ A_{k,l} &:= G(u_k)(\mathbf{z}_l), && k \in [C_U],\ l \in [C_V]. \end{aligned}$$

Set $c_P := 1 - \cos\left(\frac{2\pi}{P}\right)$ and define

$$B_{\mathrm{aff}} := 1 + (2B_{\mathcal{U}} + 3B_{\mathcal{U}}^2)M_u + 3d_2 M_y + B_{\mathcal{V}}. \tag{44}$$

For $M > 1$, define

$$\varepsilon_{\mathrm{feat}} := 2P^2 B_{\mathrm{aff}} e^{-M}.$$

If $Pe^{-M} \le \frac{1}{4}$, then there exists a first-block MHA layer $A_1 : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with

$$H^1 = (n_x + 3)P + 2, \qquad d_k^1 = 5, \qquad d_v^1 = 2,$$

such that

$$\widehat{\mathbf{Z}}_1 = A_1(\mathbf{Z}_0)$$

has the following column representations.

$$(\widehat{\mathbf{Z}}_1)_{:,j} = \begin{bmatrix} \widetilde{B}_{1,k_j,l_j} \\ \vdots \\ \widetilde{B}_{n_x,k_j,l_j} \\ \widetilde{Q}_{k_j,l_j} \\ \widetilde{N}_{k_j,l_j} \\ \widetilde{A}_{k_j,l_j} \\ \mathbf{0}_{D-n_x-7} \\ 1 \\ a_M \\ \lambda_{\mathrm{id}} \sin(\theta_j) \\ \lambda_{\mathrm{id}} \cos(\theta_j) \end{bmatrix}, \quad j \in [P-1], \qquad (\widehat{\mathbf{Z}}_1)_{:,P} = \begin{bmatrix} \delta_1^B \\ \vdots \\ \delta_{n_x}^B \\ \delta^Q \\ \delta^N \\ \delta^A \\ \mathbf{0}_{D-n_x-7} \\ 1 \\ 0 \\ \lambda_{\mathrm{id}} \sin(\theta_P) \\ \lambda_{\mathrm{id}} \cos(\theta_P) \end{bmatrix}. \tag{45}$$

The output components satisfy the following bounds.

1. ***Feature outputs.*** *For every $u \in \mathcal{U}$, $\mathbf{y} \in \Omega_{\mathcal{V}}$, and $j \in [P-1]$,*

   $$\max\left\{ \max_{i\in[n_x]} \left|\widetilde{B}_{i,k_j,l_j} - B_{i,k_j}(u)\right|, \left|\widetilde{Q}_{k_j,l_j} - Q_{l_j}(\mathbf{y})\right|, \left|\widetilde{N}_{k_j,l_j} - N_{k_j}\right|, \left|\widetilde{A}_{k_j,l_j} - A_{k_j,l_j}\right| \right\} \le \varepsilon_{\mathrm{feat}}.$$

   Moreover, the feature leakage in the null column satisfies

   $$\max\left\{ \max_{i\in[n_x]} |\delta_i^B|, |\delta^Q|, |\delta^N|, |\delta^A| \right\} \le \varepsilon_{\mathrm{feat}}.$$

2. ***Structural outputs.*** *There exist scalars $a_M, \lambda_{\mathrm{id}} > 0$ such that, for every $j \in [P]$,*

   $$(\widehat{\mathbf{Z}}_1)_{D-3,j} = 1, \qquad (\widehat{\mathbf{Z}}_1)_{D-2,j} = a_M\, 1_{\{j<P\}},$$

   and

   $$(\widehat{\mathbf{Z}}_1)_{D-1,j} = \lambda_{\mathrm{id}} \sin(\theta_j), \qquad (\widehat{\mathbf{Z}}_1)_{D,j} = \lambda_{\mathrm{id}} \cos(\theta_j).$$

   Furthermore, under $Pe^{-M} \le \frac{1}{4}$, $a_M^{-1} \le 2$ and $\lambda_{\mathrm{id}}^{-1} \le 2$.

3. ***Parameter bounds.*** *Define*

$$M_{A_1} := \max\left\{\|\mathbf{W}_1^O\|_{\max}, \max_{h\in[H^1]} \max\left\{\|\mathbf{Q}_1^h\|_{\max}, \|\mathbf{K}_1^h\|_{\max}, \|\mathbf{V}_1^h\|_{\max}\right\}\right\}.$$

Then

$$M_{A_1} \le \max\left\{1,\, B_{\text{aff}},\, \frac{5M}{c_P}\right\}.$$

Lemma 7 is proved in Appendix D.2.2.

Lemma 8 (Pair-token Joint-logit Assembly via Point-wise FFN)**.** *Let $\widehat{\mathbf{Z}}_1 \in \mathbb{R}^{D\times P}$ be the output matrix from Lemma 7. For every active column $j \in [P-1]$, the ideal joint logit can be written as*

$$\begin{aligned}\Xi_j &:= \sum_{i=1}^{n_x} B_{i,k_j}(u) - N_{k_j} + Q_{l_j}(\mathbf{y}) \\ &= 2M_u\langle u, u_{k_j}\rangle_{n_x} - M_u\|u_{k_j}\|^2_{L^2(\Omega_{\mathcal{U}})} + 2M_y\mathbf{y}^\top\mathbf{z}_{l_j} - M_y\|\mathbf{z}_{l_j}\|_2^2.\end{aligned} \tag{46}$$

Let

$$M_{\text{out}} = M + (n_x+2)\left(B_{\text{aff}} + 2\varepsilon_{\text{feat}}\right).$$

For the null column, define

$$\widehat{\Xi}_P = \sum_{i=1}^{n_x} \delta_i^B - \delta^N + \delta^Q - M_{\text{out}}.$$

There exists a point-wise FFN layer $F_1 : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with hidden width $d_{\text{ff}}^1 = 2D$ such that $\mathbf{Z}_1 = F_1(\widehat{\mathbf{Z}}_1)$ is of the form

$$\mathbf{Z}_1 = \begin{bmatrix} \widehat{\Xi}_1 & \cdots & \widehat{\Xi}_{P-1} & \widehat{\Xi}_P \\ \widetilde{A}_1 & \cdots & \widetilde{A}_{P-1} & \delta^A \\ 1 & \cdots & 1 & 1 \\ & & \mathbf{0}_{(D-3)\times P} & \end{bmatrix}, \tag{47}$$

where, for $j = j(k_j, l_j) \in [P-1]$,

$$\widehat{\Xi}_j := \sum_{i=1}^{n_x} \widetilde{B}_{i,k_j,l_j} - \widetilde{N}_{k_j,l_j} + \widetilde{Q}_{k_j,l_j}, \qquad \widetilde{A}_j := \widetilde{A}_{k_j,l_j}.$$

For every $j = j(k_j, l_j) \in [P-1]$,

$$|\widehat{\Xi}_j - \Xi_j| \le (n_x+2)\varepsilon_{\text{feat}}, \qquad |\widetilde{A}_j - A_{k_j,l_j}| \le \varepsilon_{\text{feat}}. \tag{48}$$

For the null column,

$$\widehat{\Xi}_P - \max_{j\in[P-1]} \widehat{\Xi}_j \le -M, \qquad |\delta^A| \le \varepsilon_{\text{feat}}. \tag{49}$$

Moreover, defining

$$M_{F_1} := \max\left\{\|\mathbf{W}_1^1\|_{\max}, \|\mathbf{b}_1^1\|_\infty, \|\mathbf{W}_1^2\|_{\max}, \|\mathbf{b}_1^2\|_\infty\right\},$$

the FFN parameters can be chosen so that

$$M_{F_1} \le 2M_{\text{out}}. \tag{50}$$

Lemma 8 is proved in Appendix D.2.3.

Lemma 9 (Pair-token Softmax Aggregation via MHA)**.** *Let $\mathbf{Z}_1 \in \mathbb{R}^{D\times P}$ be the output matrix from Lemma 8. For $j \in [P-1]$, define the active-only Softmax weights*

$$\widehat{\omega}_j := \frac{\exp(\widehat{\Xi}_j)}{\sum_{q=1}^{P-1}\exp(\widehat{\Xi}_q)}.$$

There exists a single-head MHA layer $A_2 : \mathbb{R}^{D\times P} \to \mathbb{R}^{D\times P}$ with $H^2 = 1$ and $d_k^2 = d_v^2 = 1$ such that $\widehat{\mathbf{Z}}_2 = A_2(\mathbf{Z}_1)$ has the form

$$\widehat{\mathbf{Z}}_2 = \begin{bmatrix} \widehat{g}(u,\mathbf{y}) & \widehat{g}(u,\mathbf{y}) & \cdots & \widehat{g}(u,\mathbf{y}) \\ & \mathbf{0}_{(D-1)\times P} & & \end{bmatrix}, \tag{51}$$

where

$$\widehat{g}(u,\mathbf{y}) = \sum_{j=1}^{P-1} \gamma_j \widetilde{A}_j + \gamma_P \delta^A,$$

and

$$\gamma_j = \frac{\exp(\widehat{\Xi}_j)}{\sum_{q=1}^{P} \exp(\widehat{\Xi}_q)}, \qquad j \in [P].$$

The attention weights satisfy

$$0 \le \gamma_P \le e^{-M},$$

and

$$\left| \widehat{g}(u,\mathbf{y}) - \sum_{j=1}^{P-1} \widehat{\omega}_j \widetilde{A}_j \right| \le (B_{\mathcal{V}} + 2\varepsilon_{\text{feat}})\, e^{-M}.$$

Moreover, defining

$$M_{A_2} := \max\left\{ \|\mathbf{W}_2^O\|_{\max}, \|\mathbf{Q}_2\|_{\max}, \|\mathbf{K}_2\|_{\max}, \|\mathbf{V}_2\|_{\max} \right\},$$

the MHA parameters can be chosen so that

$$M_{A_2} \le 1.$$

Lemma 9 is proved in Appendix D.2.4.

B.3 Proof of Theorem 1

Proof of Theorem 1. Choose $r_u := \left(\frac{\epsilon}{4(2^\gamma+1)L_G}\right)^{1/\gamma}$ and $r_y := \left(\frac{\epsilon}{4(2^\beta+1)B_{\mathcal{V}}}\right)^{1/\beta}$. By choosing ϵ_0 sufficiently small, we have $r_u, r_y \in (0,1]$. Let $\{u_k\}_{k=1}^{C_U}$ be the corresponding internal r_u-cover of $\mathcal{U}$, and let $\{\mathbf{z}_l\}_{l=1}^{C_V}$ be an r_y-cover of $\Omega_{\mathcal{V}}$. Set $P := C_U C_V + 1$ and define M_u and M_y as in (32) and (36).

Choose $n_x := \left\lceil \left(\frac{16 B_{\mathcal{V}} M_u C_{\text{quad}}}{\epsilon}\right)^{d_1/\alpha} \right\rceil$. By taking ϵ_0 sufficiently small, we have $n_x \ge (2s_{\text{quad}})^{d_1}$, so Lemma 3 applies. We also use the sequence-length compatibility condition $P \ge n_x + 2$. Then Lemma 5 gives

$$\sup_{u\in\mathcal{U}} \left\| G(u) - \widetilde{G}_{r_u,r_y,n_x}(u) \right\|_{L^2(\rho_y)} \le \frac{3\epsilon}{4}. \tag{52}$$

Transformer construction. For each $u \in \mathcal{U}$ and $\mathbf{y} \in \Omega_{\mathcal{V}}$, Lemma 6 constructs the initial sequence matrix

$$\mathbf{Z}_0 = \mathcal{P}(S_{n_x}(u), \mathbf{y}) \in \mathbb{R}^{D\times P},$$

with the structure in (43). The first n_x columns store the discretized input information, the (n_x+1)-st column stores the query point $\mathbf{y}$, and the last column is reserved as the null column.

For Encoder Block 1, Lemma 7 first constructs

$$\widehat{\mathbf{Z}}_1 = A_1(\mathbf{Z}_0),$$

whose active columns contain approximations of the target features $B_{i,k}(u)$, $Q_l(\mathbf{y})$, N_k, and $A_{k,l}$.

Applying Lemma 8 then gives

$$\mathbf{Z}_1 = F_1(\widehat{\mathbf{Z}}_1),$$

with the form in (47). In particular, for each active column $j = j(k,l)$, the first two coordinates are the assembled joint logit $\widehat{\Xi}_j$ and the corresponding anchor value $\widetilde{A}_j$, respectively. The null column satisfies the separation property (49).

For Encoder Block 2, applying Lemma 9 gives a single-head MHA layer $\widehat{\mathbf{Z}}_2 = A_2(\mathbf{Z}_1)$ such that the output has the form (51). In particular, the first coordinate of every column equals

$$\widehat{g}(u,\mathbf{y}) = \sum_{j=1}^{P-1} \gamma_j \widetilde{A}_j + \gamma_P \delta^A,$$

where $\widetilde{A}_j = \widetilde{A}_{k_j,l_j}$ and $\gamma_1, \ldots, \gamma_P$ are the attention weights constructed in Lemma 9.

The second point-wise FFN is chosen as the identity map. Specifically, set

$$d_{\text{ff}}^2 = 2D, \qquad \mathbf{W}_2^1 = \begin{bmatrix} I_D \\ -I_D \end{bmatrix}, \qquad \mathbf{b}_2^1 = \mathbf{0}_{2D},$$

$$\mathbf{W}_2^2 = [I_D \quad -I_D], \qquad \mathbf{b}_2^2 = \mathbf{0}_D.$$

Using $x = \sigma(x) - \sigma(-x)$ componentwise,

$$\mathbf{Z}_2 = F_2(\widehat{\mathbf{Z}}_2) = \widehat{\mathbf{Z}}_2.$$

Finally, the scalar output is obtained by the linear readout $\mathbf{c}^\top \operatorname{vec}(\mathbf{Z}_2)$, where $\mathbf{c} \in \mathbb{R}^{DP}$ is the first standard basis vector. Hence

$$\mathbf{c}^\top \operatorname{vec}(\mathbf{Z}_2) = (\mathbf{Z}_2)_{1,1} = \widehat{g}(u,\mathbf{y}).$$

Error bound analysis. For an active column $j = j(k,l) \in [P-1]$, let Ξ_j be the ideal joint logit defined in (46). Since it is the sum of the input-space and output-domain logits,

$$\frac{\exp(\Xi_j)}{\sum_{q=1}^{P-1} \exp(\Xi_q)} = \widetilde{\beta}_{k,n_x}(u)\widetilde{\eta}_l(\mathbf{y}).$$

By Lemma 8,

$$\max_{j\in[P-1]} |\widehat{\Xi}_j - \Xi_j| \le (n_x+2)\varepsilon_{\text{feat}},$$

and

$$|\widetilde{A}_j - A_{k_j,l_j}| \le \varepsilon_{\text{feat}}.$$

Hence Lemma 4 gives

$$\sum_{j=1}^{P-1} \left| \widehat{\omega}_j - \widetilde{\beta}_{k_j,n_x}(u)\widetilde{\eta}_{l_j}(\mathbf{y}) \right| \le 2(n_x+2)\varepsilon_{\text{feat}}.$$

Using $|\widetilde{A}_j - A_{k_j,l_j}| \le \varepsilon_{\text{feat}}$, $|A_{k_j,l_j}| \le B_{\mathcal{V}}$ and (23), we obtain

$$\left| \sum_{j=1}^{P-1} \widehat{\omega}_j \widetilde{A}_j - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y}) \right| \le \varepsilon_{\text{feat}} + 2B_{\mathcal{V}}(n_x+2)\varepsilon_{\text{feat}} = (1 + 2B_{\mathcal{V}}(n_x+2))\,\varepsilon_{\text{feat}}.$$

Combining this with Lemma 9 yields

$$\left| \widehat{g}(u,\mathbf{y}) - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y}) \right| \le (1 + 2B_{\mathcal{V}}(n_x+2))\,\varepsilon_{\text{feat}} + (B_{\mathcal{V}} + 2\varepsilon_{\text{feat}})\,e^{-M}.$$

Since $\varepsilon_{\text{feat}} = 2P^2 B_{\text{aff}} e^{-M}$ and $M > 1$, it follows that

$$\left| \widehat{g}(u,\mathbf{y}) - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y}) \right| \le \left[B_{\mathcal{V}} + 2P^2 B_{\text{aff}}\,(3 + 2B_{\mathcal{V}}(n_x+2))\right] e^{-M}.$$

Choose $M := \log\left(\frac{4\left[B_{\mathcal{V}} + 2P^2 B_{\text{aff}}(3+2B_{\mathcal{V}}(n_x+2))\right]}{\epsilon} \right)$. Then

$$\left| \widehat{g}(u,\mathbf{y}) - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y}) \right| \le \frac{\epsilon}{4}.$$

Since the preceding bound holds uniformly for $\mathbf{y} \in \Omega_{\mathcal{V}}$ and ρ_y is a probability distribution,

$$\left\| \widehat{g}(u,\cdot) - \widetilde{G}_{r_u,r_y,n_x}(u) \right\|_{L^2(\rho_y)} \le \frac{\epsilon}{4}.$$

Moreover, since $B_{\text{aff}} \ge 1$, $P \ge 1$, and $\epsilon \le 1$, this choice also gives $Pe^{-M} \le \frac{1}{4}$, so the conditions of the preceding Transformer construction lemmas are satisfied.

Combining the implementation error with (52) and using the triangle inequality in $L^2(\rho_y)$ gives

$$\begin{aligned}\sup_{u\in\mathcal{U}} \|T_\Theta(S_{n_x}(u),\cdot) - G(u)\|_{L^2(\rho_y)} &\le \sup_{u\in\mathcal{U}} \left\| T_\Theta(S_{n_x}(u),\cdot) - \widetilde{G}_{r_u,r_y,n_x}(u) \right\|_{L^2(\rho_y)} \\ &\quad + \sup_{u\in\mathcal{U}} \left\| \widetilde{G}_{r_u,r_y,n_x}(u) - G(u) \right\|_{L^2(\rho_y)} \\ &\le \frac{\epsilon}{4} + \frac{3\epsilon}{4} = \epsilon.\end{aligned}$$

Parameter complexity. We count the total architectural (dense) parameters N_{total} of the constructed Transformer:

- **Pre-processing:** The shared affine embedding, its bias, and the fixed structural-positional matrix have $D(P + d_1 + d_2 + 2)$ parameters.
- **MHA$_1$:** With $H^1 = (n_x+3)P + 2$, $d_k^1 = 5$, and $d_v^1 = 2$, there are $14D\big((n_x+3)P+2\big)$ parameters.
- **FFN$_1$:** Hidden dimension $2D$ gives $4D^2 + 3D$ parameters.
- **MHA$_2$:** One head with $d_k^2 = d_v^2 = 1$ gives $4D$ parameters.
- **FFN$_2$:** Hidden dimension $2D$ gives $4D^2 + 3D$ parameters.
- **Readout:** The linear projection over $\mathrm{vec}(\mathbf{Z}_2)$ has DP parameters.

Summing these contributions yields

$$\begin{aligned}N_{\text{total}} &= D(P + d_1 + d_2 + 2) + 14D\big((n_x+3)P+2\big) + 2(4D^2+3D) + 4D + DP \\ &= 8D^2 + 14D(n_x+3)P + D(d_1 + d_2 + 2P + 2) + 38D.\end{aligned}$$

Since $n_x \ge 1$, $P \ge 1$, and $D = d_1 + d_2 + n_x + 9 \le (d_1 + d_2 + 10)n_x$, we obtain

$$\begin{aligned}N_{\text{total}} &\le 8(d_1+d_2+10)^2 n_x^2 P + 56(d_1+d_2+10)n_x^2 P + (d_1+d_2+10)(d_1+d_2+4)n_x^2 P + 38(d_1+d_2+10)n_x^2 P \\ &= (d_1+d_2+10)(9d_1+9d_2+178)n_x^2 P \\ &=: C_{\text{arch}} n_x^2 P.\end{aligned}$$

where $C_{\text{arch}} := (d_1+d_2+10)(9d_1+9d_2+178)$. By (12) and the choices of r_u and r_y,

$$C_U \le C_1 \left(4(2^\gamma+1)L_G\right)^{d_{\mathcal{U}}/\gamma} \epsilon^{-d_{\mathcal{U}}/\gamma}.$$

Since $\Omega_{\mathcal{V}} \subseteq [0,1]^{d_2}$ is compact, it admits an internal r_y-cover with

$$C_V \le C_2 r_y^{-d_2},$$

where $C_2 > 0$ depends only on d_2. Therefore, by the choice of r_y,

$$C_V \le C_2 \left(4(2^\beta+1)B_{\mathcal{V}}\right)^{d_2/\beta} \epsilon^{-d_2/\beta}.$$

for a constant $C_2 > 0$ depending only on d_2. Hence

$$P = C_U C_V + 1 \le C_P \epsilon^{-d_{\mathcal{U}}/\gamma - d_2/\beta}, \tag{53}$$

where

$$C_P := 1 + C_1 C_2 \left(4(2^\gamma+1)L_G\right)^{d_{\mathcal{U}}/\gamma} \left(4(2^\beta+1)B_{\mathcal{V}}\right)^{d_2/\beta}. \tag{54}$$

By (32), (12), and the choice of r_u, we have

$$M_u \le \frac{1}{3}\left(\frac{4(2^\gamma+1)L_G}{\epsilon}\right)^{2/\gamma} \log\left[\frac{2B_{\mathcal{V}}C_1}{L_G}\left(\frac{4(2^\gamma+1)L_G}{\epsilon}\right)^{(d_{\mathcal{U}}+\gamma)/\gamma}\right].$$

Hence, for $\epsilon \le e^{-1}$,

$$M_u \le C_{M_u}\epsilon^{-2/\gamma}\log\frac{1}{\epsilon}, \tag{55}$$

where

$$C_{M_u} = \frac{(4(2^\gamma+1)L_G)^{2/\gamma}}{3}\left[\max\left\{\log\left(\frac{2B_{\mathcal{V}}C_1}{L_G}\left(4(2^\gamma+1)L_G\right)^{(d_{\mathcal{U}}+\gamma)/\gamma}\right), 0\right\} + 1 + \frac{d_{\mathcal{U}}}{\gamma}\right].$$

Therefore, by the choice of n_x,

$$n_x \le C_x\epsilon^{-\frac{d_1}{\alpha}(1+\frac{2}{\gamma})}\left(\log\frac{1}{\epsilon}\right)^{d_1/\alpha}, \tag{56}$$

where

$$C_x := 2\left(16B_{\mathcal{V}}C_{\mathrm{quad}}C_{M_u}\right)^{d_1/\alpha}. \tag{57}$$

Combining (53) and (56) with $N_{\mathrm{total}} \le C_{\mathrm{arch}}n_x^2 P$ gives

$$N_{\mathrm{total}} \le C_N\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{2d_1/\alpha},$$

where

$$C_N := C_{\mathrm{arch}}C_x^2C_P. \tag{58}$$

Parameter magnitude. By Lemmas 7, 8, and 9,

$$M_{A_1} \le \max\left\{1,\, B_{\mathrm{aff}},\, \frac{5M}{c_P}\right\}.$$

$$M_{F_1} \le 2M_{\mathrm{out}} = 2M + 2(n_x+2)\left(B_{\mathrm{aff}} + 2\varepsilon_{\mathrm{feat}}\right), \qquad M_{A_2} \le 1.$$

The preprocessing parameters, the second identity FFN, and the final readout have magnitude at most 1. By the choice of M in the error-bound analysis,

$$\varepsilon_{\mathrm{feat}} \le \frac{\epsilon}{12} \le \frac{1}{12}.$$

Since $B_{\mathrm{aff}} \ge 1$,

$$B_{\mathrm{aff}} + 2\varepsilon_{\mathrm{feat}} \le B_{\mathrm{aff}} + \frac{1}{6} \le \frac{7}{6}B_{\mathrm{aff}}.$$

Using $n_x \ge 1$, so that $n_x + 2 \le 3n_x$, we obtain

$$\begin{aligned} M_{F_1} &\le 2M + 2(n_x+2)\left(B_{\mathrm{aff}} + 2\varepsilon_{\mathrm{feat}}\right) \\ &\le 2M + 2(3n_x)\frac{7}{6}B_{\mathrm{aff}} \\ &= 2M + 7n_xB_{\mathrm{aff}}. \end{aligned}$$

Moreover,

$$c_P = 2\sin^2\left(\frac{\pi}{P}\right) \le 2,$$

and hence

$$M \le 2\frac{M}{c_P}.$$

Therefore,

$$M_{F_1} \le 4\frac{M}{c_P} + 7n_xB_{\mathrm{aff}}.$$

Since $n_x \ge 1$ and $B_{\mathrm{aff}} \ge 1$,

$$M_{A_1} \le 5\max\left\{\frac{M}{c_P}, n_xB_{\mathrm{aff}}\right\}.$$

Consequently,

$$M_{\max} \le 11 \max\left\{1, \frac{M}{c_P}, n_x B_{\text{aff}}\right\}.$$

The elementary bound $\sin(\pi/P) \ge 2/P$ for $P \ge 2$ gives

$$\frac{1}{c_P} \le \frac{P^2}{8}. \tag{59}$$

Moreover, by (36), the choice of r_y, and $C_V \le C_2 r_y^{-d_2}$, for $\epsilon \le e^{-1}$,

$$M_y \le C_{M_y} \epsilon^{-2/\beta} \log\frac{1}{\epsilon}, \tag{60}$$

where

$$C_{M_y} = \frac{\left(4(2^\beta+1)B_{\mathcal{V}}\right)^{2/\beta}}{3}\left[\max\left\{\log\left(2C_2\left(4(2^\beta+1)B_{\mathcal{V}}\right)^{(d_2+\beta)/\beta}\right), 0\right\} + 1 + \frac{d_2}{\beta}\right].$$

By (44), (55), and (60), define

$$C_{\text{aff}} := 1 + B_{\mathcal{V}} + (2B_{\mathcal{U}} + 3B_{\mathcal{U}}^2)C_{M_u} + 3d_2 C_{M_y}.$$

Then, since $\epsilon \le e^{-1}$,

$$B_{\text{aff}} \le C_{\text{aff}}\left(\epsilon^{-2/\gamma} + \epsilon^{-2/\beta}\right)\log\frac{1}{\epsilon}. \tag{61}$$

We next bound the routing parameter M. Set

$$C_{\text{aux}} := B_{\mathcal{V}} + 4C_P^2 C_{\text{aff}} C_x\left(3 + 6B_{\mathcal{V}}\right).$$

By the choice of M in the error-bound analysis, (53), (56), and (61), using $n_x + 2 \le 3n_x$, we obtain

$$B_{\mathcal{V}} + 2P^2 B_{\text{aff}}\left(3 + 2B_{\mathcal{V}}(n_x+2)\right) \le C_{\text{aux}}\epsilon^{-R_M}\left(\log\frac{1}{\epsilon}\right)^{1+d_1/\alpha},$$

where

$$R_M := 2\left(\frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}\right) + \frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + \max\left\{\frac{2}{\gamma}, \frac{2}{\beta}\right\}.$$

Therefore,

$$M \le C_M \log\frac{1}{\epsilon},$$

where one may take

$$C_M := \max\left\{1,\ \max\left\{\log(4C_{\text{aux}}), 0\right\} + R_M + \frac{d_1}{\alpha} + 2\right\}.$$

Combining this bound with (59) and (53) gives

$$\frac{M}{c_P} \le \frac{C_M C_P^2}{8}\epsilon^{-2\left(\frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}\right)}\log\frac{1}{\epsilon}.$$

Moreover, by (56) and (61),

$$n_x B_{\text{aff}} \le C_x C_{\text{aff}}\left[\epsilon^{-\frac{d_1}{\alpha}(1+\frac{2}{\gamma})-\frac{2}{\gamma}} + \epsilon^{-\frac{d_1}{\alpha}(1+\frac{2}{\gamma})-\frac{2}{\beta}}\right]\left(\log\frac{1}{\epsilon}\right)^{1+d_1/\alpha}.$$

Therefore,

$$M_{\max} \le C_{\text{mag}}\epsilon^{-q_M}\left(\log\frac{1}{\epsilon}\right)^{1+d_1/\alpha},$$

where

$$q_M := \max\left\{2\left(\frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}\right), \frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + \frac{2}{\gamma}, \frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + \frac{2}{\beta}\right\},$$

and

$$C_{\text{mag}} := 11 \max\left\{1, \frac{C_M C_P^2}{8}, 2C_x C_{\text{aff}}\right\}. \tag{62}$$

□

C Proof of Theorem 2: Generalization and Data Scaling

Following the proof sketch in Section 4, Appendix C.1 establishes the covering-number bound for the Transformer class, which controls the variance term. Appendix C.2 then uses the approximation bound from Theorem 1 to control the bias term, combines the resulting bias and variance bounds, and balances them to prove Theorem 2 and derive the data scaling law.

C.1 Covering Number Bound for the Transformer Class

Lemma 10 (Covering Number of the Transformer Class)**.** *Let $\mathcal{T}_\epsilon$ denote the class of two-block Transformers with the architecture and parameter bounds specified in Theorem 1, with fixed preprocessing as in Lemma 6, and let $\mathcal{H}_\epsilon^{\mathrm{clip}}$ be the corresponding clipped class. Assume $P \geq n_x + 2$ and $M_{\max} \geq 1$. Then, for every $\eta \in (0, 1]$,*

$$\log \mathcal{N}\left(\eta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right) \leq N_{\mathrm{total}} \log\left(\frac{503552\, P^7 D^{17} M_{\max}^{21} B_{\mathcal{U}}^5}{\eta}\right). \tag{63}$$

The proof of Lemma 10 is given in Appendix D.3.

C.2 Proof of Theorem 2

Proof of Theorem 2. By Theorem 1, there exists $T_\epsilon^* \in \mathcal{T}_\epsilon$ such that

$$\sup_{u\in\mathcal{U}} \|T_\epsilon^*(S_{n_x}(u), \cdot) - G(u)\|_{L^2(\rho_y)} \leq \epsilon.$$

Since $|G(u)(\mathbf{y})| \leq B_{\mathcal{V}}$ for every $u \in \mathcal{U}$ and $\mathbf{y} \in \Omega_{\mathcal{V}}$, clipping does not increase the pointwise approximation error. Hence $\Pi_{B_{\mathcal{V}}} \circ T_\epsilon^* \in \mathcal{H}_\epsilon^{\mathrm{clip}}$ and

$$\mathcal{L}(\Pi_{B_{\mathcal{V}}} \circ T_\epsilon^*) \leq \epsilon^2.$$

For simplicity, we continue to denote this clipped comparator by T_ϵ^*.

Define the empirical clean error by

$$\|T - G\|_{\mathcal{S}}^2 := \frac{1}{nn_y} \sum_{i=1}^{n} \sum_{j=1}^{n_y} \left(T(S_{n_x}(u_i), \mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)^2.$$

Following the decomposition in the proof of Theorem 2 of Liu et al. (2026), write

$$\mathbb{E}_{\mathcal{S}} \mathcal{L}(\widehat{T}_\epsilon) = T_1 + T_2,$$

where

$$T_1 := 2\mathbb{E}_{\mathcal{S}} \|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2,$$

and

$$T_2 := \mathbb{E}_{\mathcal{S}} \mathcal{L}(\widehat{T}_\epsilon) - 2\mathbb{E}_{\mathcal{S}} \|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2.$$

Bounding T_1. Recall that

$$v_{i,j} = G(u_i)(\mathbf{y}_{i,j}) + \xi_{i,j}.$$

By the ERM property of $\widehat{T}_\epsilon$,

$$\frac{1}{nn_y} \sum_{i=1}^{n} \sum_{j=1}^{n_y} \left(\widehat{T}_\epsilon(S_{n_x}(u_i), \mathbf{y}_{i,j}) - v_{i,j}\right)^2 \leq \frac{1}{nn_y} \sum_{i=1}^{n} \sum_{j=1}^{n_y} \left(T_\epsilon^*(S_{n_x}(u_i), \mathbf{y}_{i,j}) - v_{i,j}\right)^2.$$

Expanding $v_{i,j} = G(u_i)(\mathbf{y}_{i,j}) + \xi_{i,j}$ gives

$$\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2 \leq \|T_\epsilon^* - G\|_{\mathcal{S}}^2 + \frac{2}{nn_y} \sum_{i=1}^{n} \sum_{j=1}^{n_y} \left(\widehat{T}_\epsilon(S_{n_x}(u_i), \mathbf{y}_{i,j}) - T_\epsilon^*(S_{n_x}(u_i), \mathbf{y}_{i,j})\right) \xi_{i,j}.$$

Writing

$$\widehat{T}_\epsilon - T_\epsilon^* = (\widehat{T}_\epsilon - G) - (T_\epsilon^* - G),$$

the preceding inequality becomes

$$\begin{aligned}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2 \le \|T_\epsilon^* - G\|_{\mathcal{S}}^2 &\\ + \frac{2}{nn_y}\sum_{i=1}^{n}\sum_{j=1}^{n_y}\left(\widehat{T}_\epsilon(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\xi_{i,j} &\\ - \frac{2}{nn_y}\sum_{i=1}^{n}\sum_{j=1}^{n_y}\left(T_\epsilon^*(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\xi_{i,j}.&\end{aligned}$$

Since T_ϵ^* is fixed independently of the training sample and $\mathbb{E}[\xi_{i,j} \mid u_i, \mathbf{y}_{i,j}] = 0$,

$$\begin{aligned}&\mathbb{E}_{\mathcal{S}}\left[\left(T_\epsilon^*(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\xi_{i,j}\right]\\ &\quad= \mathbb{E}_{u_i,\mathbf{y}_{i,j}}\left[\left(T_\epsilon^*(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\mathbb{E}[\xi_{i,j} \mid u_i, \mathbf{y}_{i,j}]\right] = 0.\end{aligned}$$

Moreover,

$$\mathbb{E}_{\mathcal{S}}\|T_\epsilon^* - G\|_{\mathcal{S}}^2 = \mathcal{L}(T_\epsilon^*) \le \epsilon^2.$$

Therefore,

$$\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2 \le \epsilon^2 + \frac{2}{nn_y}\mathbb{E}_{\mathcal{S}}\sum_{i=1}^{n}\sum_{j=1}^{n_y}\left(\widehat{T}_\epsilon(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\xi_{i,j}.$$

The argument of Lemma 3 in Liu et al. (2026) applies to $\mathcal{H}_\epsilon^{\mathrm{clip}}$ under our sampling and noise assumptions. Thus, for every $\theta > 0$,

$$\begin{aligned}&\frac{1}{nn_y}\mathbb{E}_{\mathcal{S}}\sum_{i=1}^{n}\sum_{j=1}^{n_y}\left(\widehat{T}_\epsilon(S_{n_x}(u_i),\mathbf{y}_{i,j}) - G(u_i)(\mathbf{y}_{i,j})\right)\xi_{i,j}\\ &\qquad\le 2\sigma\left(\sqrt{\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2} + \theta\right)\sqrt{\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}} + \sigma\theta.\end{aligned}$$

Consequently,

$$T_1 = 2\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2 \le 2\epsilon^2 + 8\sigma\left(\sqrt{\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2} + \theta\right)\sqrt{\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}} + 4\sigma\theta.$$

Denote

$$\eta := \sqrt{\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2},$$

and set

$$a := \epsilon^2 + 2\sigma\theta + 4\sigma\theta\sqrt{\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}},$$

$$b := 2\sigma\sqrt{\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}}.$$

Then

$$\eta^2 \le a + 2b\eta.$$

Hence

$$(\eta - b)^2 \le a + b^2,$$

which implies

$$\eta^2 \le 2a + 4b^2.$$

Therefore,

$$\begin{aligned}&T_1 = 2\mathbb{E}_{\mathcal{S}}\|\widehat{T}_\epsilon - G\|_{\mathcal{S}}^2\\ &\quad\le 4\epsilon^2 + 8\sigma\theta + 16\sigma\theta\sqrt{\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}} + 32\sigma^2\frac{4\log\mathcal{N}\left(\theta,\mathcal{H}_\epsilon^{\mathrm{clip}},\|\cdot\|_{\infty,\infty}\right)+6}{nn_y}.\end{aligned}$$

Bounding T_2. The argument of Lemma 4 in Liu et al. (2026) applies to the clipped Transformer class $\mathcal{H}_\epsilon^{\text{clip}}$ under our sampling and boundedness assumptions. Thus, for every $\theta > 0$,

$$T_2 \le \frac{19B_{\mathcal{V}}^2}{n} \log \mathcal{N}\left(\frac{\theta}{4B_{\mathcal{V}}}, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) + 6\theta.$$

Putting T_1 and T_2 together. Combining the bounds for T_1 and T_2, we obtain

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T}_\epsilon) &= T_1 + T_2 \\
&\le 4\epsilon^2 + 8\sigma\theta \\
&\quad + 16\sigma\theta\sqrt{\frac{4\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) + 6}{nn_y}} \\
&\quad + 32\sigma^2\frac{4\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) + 6}{nn_y} \\
&\quad + \frac{19B_{\mathcal{V}}^2}{n}\log\mathcal{N}\left(\frac{\theta}{4B_{\mathcal{V}}}, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) + 6\theta.
\end{aligned}$$

By Lemma 10,

$$\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) \le N_{\text{total}}\log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right),$$

and

$$\log\mathcal{N}\left(\frac{\theta}{4B_{\mathcal{V}}}, \mathcal{H}_\epsilon^{\text{clip}}, \|\cdot\|_{\infty,\infty}\right) \le N_{\text{total}}\log\left(\frac{2014208B_{\mathcal{V}}P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right).$$

Therefore,

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T}_\epsilon) &\le 4\epsilon^2 + (8\sigma+6)\theta \\
&\quad + 16\sigma\theta\sqrt{\frac{4N_{\text{total}}\log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right) + 6}{nn_y}} \\
&\quad + 32\sigma^2\frac{4N_{\text{total}}\log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right) + 6}{nn_y} \\
&\quad + \frac{19B_{\mathcal{V}}^2N_{\text{total}}}{n}\log\left(\frac{2014208B_{\mathcal{V}}P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right).
\end{aligned}$$

We now use the covering-number estimate from Lemma 10. Choose

$$\theta := \frac{\epsilon^2}{1+\sigma}.$$

By decreasing ϵ_0 if necessary, we may assume that both θ and $\theta/(4B_{\mathcal{V}})$ belong to $(0, 1]$.

By Theorem 1,

$$P \le C_P\epsilon^{-\frac{d_{\mathcal{U}}}{\gamma}-\frac{d_2}{\beta}},$$

$$D \le (d_1+d_2+10)C_x\epsilon^{-\frac{d_1}{\alpha}(1+\frac{2}{\gamma})}\left(\log\frac{1}{\epsilon}\right)^{d_1/\alpha},$$

and

$$M_{\max} \le C_{\text{mag}}\epsilon^{-q_M}\left(\log\frac{1}{\epsilon}\right)^{1+d_1/\alpha}.$$

Define

$$C_{\text{cov}} := 2014208(1+\sigma)(1+B_{\mathcal{V}})B_{\mathcal{U}}^5C_P^7\left((d_1+d_2+10)C_x\right)^{17}C_{\text{mag}}^{21},$$

and

$$C_{\log} := 1 + |\log C_{\mathrm{cov}}| + 23 + 7\left(\frac{d_{\mathcal{U}}}{\gamma} + \frac{d_2}{\beta}\right) + 17\frac{d_1}{\alpha}\left(1 + \frac{2}{\gamma}\right) + 21q_M + 38\frac{d_1}{\alpha}.$$

Since $\epsilon \le e^{-1}$, we have $\log\log(1/\epsilon) \le \log(1/\epsilon)$. Hence

$$\log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right) \le C_{\log}\log\frac{1}{\epsilon},$$

and

$$\log\left(\frac{2014208B_{\mathcal{V}}P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\theta}\right) \le C_{\log}\log\frac{1}{\epsilon}.$$

Therefore, by Lemma 10 and the parameter-count bound in Theorem 1,

$$\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right) \le C_N C_{\log}\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1},$$

and

$$\log\mathcal{N}\left(\frac{\theta}{4B_{\mathcal{V}}}, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right) \le C_N C_{\log}\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1}.$$

Moreover,

$$(8\sigma+6)\theta = \frac{8\sigma+6}{1+\sigma}\epsilon^2 \le 8\epsilon^2.$$

Using $2ab \le a^2 + b^2$,

$$\begin{aligned}
&16\sigma\theta\sqrt{\frac{4\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right)+6}{nn_y}} \\
&\quad\le 8\theta + 8\sigma^2\theta\frac{4\log\mathcal{N}\left(\theta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right)+6}{nn_y} \\
&\quad\le 8\epsilon^2 + 8\sigma^2\frac{4C_NC_{\log}\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1}+6}{nn_y}.
\end{aligned}$$

Since $\epsilon \le e^{-1}$,

$$\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1} \ge 1.$$

Substituting the preceding estimates into the bound for $T_1 + T_2$ gives

$$\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T}_\epsilon) \le C_{\mathrm{gen}}\left[\epsilon^2 + \left(1 + \frac{\sigma^2}{n_y}\right)\frac{\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1}}{n}\right],$$

where one may take

$$C_{\mathrm{gen}} := 20 + 40\left(4C_NC_{\log}+6\right) + 19B_{\mathcal{V}}^2C_NC_{\log}. \tag{64}$$

This proves the generalization bound.

To obtain the stated statistical rate, choose

$$\epsilon := \left[\frac{1+\sigma^2/n_y}{n}(\log n)^{\frac{2d_1}{\alpha}+1}\right]^{\frac{1}{2+s}}.$$

For any $n \ge 2$ such that this choice satisfies $\epsilon \in (0, \epsilon_0)$ and the sequence-length compatibility condition $P \ge n_x + 2$, since $s \ge 2d_1/\alpha$ and $1+\sigma^2/n_y \ge 1$, we have $\epsilon \ge n^{-1}$. Thus, $\log\frac{1}{\epsilon} \le \log n$.

Moreover, by definition of ϵ,

$$\epsilon^{2+s} = \frac{1+\sigma^2/n_y}{n}(\log n)^{\frac{2d_1}{\alpha}+1}.$$

Therefore,

$$\left(1+\frac{\sigma^2}{n_y}\right)\frac{\epsilon^{-s}\left(\log\frac{1}{\epsilon}\right)^{\frac{2d_1}{\alpha}+1}}{n} \le \left(1+\frac{\sigma^2}{n_y}\right)\frac{\epsilon^{-s}(\log n)^{\frac{2d_1}{\alpha}+1}}{n} = \epsilon^2.$$

Hence

$$\mathbb{E}_{\mathcal{S}}\mathcal{L}(\widehat{T}_\epsilon) \le 2C_{\mathrm{gen}}\epsilon^2 = C_{\mathrm{rate}}\left[\frac{1+\sigma^2/n_y}{n}(\log n)^{\frac{2d_1}{\alpha}+1}\right]^{\frac{2}{2+s}},$$

where

$$C_{\mathrm{rate}} := 2C_{\mathrm{gen}}.$$

This completes the proof. □

D Proofs of Supporting Lemmas

D.1 Proofs for the Two-Level Softmax POU Approximation

D.1.1 Proof of Lemma 1

Proof of Lemma 1. Fix an arbitrary $u \in \mathcal{U}$.

Step 1: Covering and definition of the input-space Softmax POU. Since $\{u_k\}_{k=1}^{C_U} \subset \mathcal{U}$ is an internal r_u-cover of $\mathcal{U}$ under the $L^2(\Omega_{\mathcal{U}})$ metric, we can choose $k^*(u) \in \arg\min_{k\in[C_U]} \|u-u_k\|_{L^2(\Omega_{\mathcal{U}})}$ such that $\|u-u_{k^*(u)}\|_{L^2(\Omega_{\mathcal{U}})} \le r_u$. The input-space weights are defined in (33) using affine logits. Equivalently, they can be written in the distance-score form

$$\widetilde{\beta}_k(u) = \frac{\exp\left(M_u\left(r_u^2 - \|u-u_k\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right)}{\sum_{k'=1}^{C_U}\exp\left(M_u\left(r_u^2 - \|u-u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right)}, \qquad k\in[C_U]. \tag{65}$$

Then $\widetilde{\beta}_k(u) \ge 0$ and $\sum_{k=1}^{C_U}\widetilde{\beta}_k(u) = 1$.

To see the equivalence with (33), notice that $r_u^2 - \|u\|^2_{L^2(\Omega_{\mathcal{U}})}$ is common to all Softmax logits and therefore cancels in the normalization. Indeed,

$$\begin{aligned}\widetilde{\beta}_k(u) &= \frac{\exp\left(M_u\left(r_u^2-\|u\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right)\exp\left(2M_u\langle u,u_k\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_k\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}{\sum_{k'=1}^{C_U}\exp\left(M_u\left(r_u^2-\|u\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right)\exp\left(2M_u\langle u,u_{k'}\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)} \\ &= \frac{\exp\left(2M_u\langle u,u_k\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_k\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}{\sum_{k'=1}^{C_U}\exp\left(2M_u\langle u,u_{k'}\rangle_{L^2(\Omega_{\mathcal{U}})} - M_u\|u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)}.\end{aligned}$$

Step 2: Near and far error estimates. Using the partition-of-unity property, for every $u \in \mathcal{U}$,

$$G(u) - G_{r_u}(u) = G(u) - \sum_{k=1}^{C_U}\widetilde{\beta}_k(u)G(u_k) = \sum_{k=1}^{C_U}\widetilde{\beta}_k(u)\big(G(u)-G(u_k)\big).$$

By the triangle inequality in $L^2(\rho_y)$,

$$\begin{aligned}\|G(u)-G_{r_u}(u)\|_{L^2(\rho_y)} &\le \sum_{k=1}^{C_U}\widetilde{\beta}_k(u)\|G(u)-G(u_k)\|_{L^2(\rho_y)} \\ &= \underbrace{\sum_{k:\,\|u-u_k\|_{L^2(\Omega_{\mathcal{U}})}\le 2r_u}\widetilde{\beta}_k(u)\|G(u)-G(u_k)\|_{L^2(\rho_y)}}_{(I)} \\ &\quad + \underbrace{\sum_{k:\,\|u-u_k\|_{L^2(\Omega_{\mathcal{U}})}> 2r_u}\widetilde{\beta}_k(u)\|G(u)-G(u_k)\|_{L^2(\rho_y)}}_{(II)}.\end{aligned}$$

For the near-center term (I), Assumption 1 yields

$$\|G(u) - G(u_k)\|_{L^2(\rho_y)} \le L_G \|u - u_k\|^{\gamma}_{L^2(\Omega_{\mathcal{U}})} \le L_G (2r_u)^{\gamma}$$

for all k satisfying $\|u - u_k\|_{L^2(\Omega_{\mathcal{U}})} \le 2r_u$. Therefore,

$$(I) \le 2^{\gamma} L_G r_u^{\gamma}.$$

For the far-center term (II), since ρ_y is a probability distribution and Assumption 3 gives $\|G(u)\|_{L^\infty(\Omega_{\mathcal{V}})} \le B_{\mathcal{V}}$ for every $u \in \mathcal{U}$, we have

$$\begin{aligned}\|G(u) - G(u_k)\|_{L^2(\rho_y)} &\le \|G(u)\|_{L^2(\rho_y)} + \|G(u_k)\|_{L^2(\rho_y)} \\ &\le \|G(u)\|_{L^\infty(\Omega_{\mathcal{V}})} + \|G(u_k)\|_{L^\infty(\Omega_{\mathcal{V}})} \\ &\le 2B_{\mathcal{V}}.\end{aligned}$$

It remains to bound the total Softmax mass assigned to far centers. By the distance-score representation in (65), the denominator satisfies

$$\sum_{k'=1}^{C_U} \exp\left(M_u \left(r_u^2 - \|u - u_{k'}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right) \ge \exp\left(M_u \left(r_u^2 - \|u - u_{k^*(u)}\|^2_{L^2(\Omega_{\mathcal{U}})}\right)\right) \ge 1.$$

If $\|u - u_k\|_{L^2(\Omega_{\mathcal{U}})} > 2r_u$, then

$$r_u^2 - \|u - u_k\|^2_{L^2(\Omega_{\mathcal{U}})} < r_u^2 - 4r_u^2 = -3r_u^2.$$

So, $\widetilde{\beta}_k(u) \le \exp(-3M_u r_u^2)$. Then

$$\sum_{k:\, \|u-u_k\|_{L^2(\Omega_{\mathcal{U}})} > 2r_u} \widetilde{\beta}_k(u) \le C_U e^{-3M_u r_u^2}.$$

Therefore,

$$(II) \le 2B_{\mathcal{V}} C_U e^{-3M_u r_u^2}.$$

Combining the estimates for (I) and (II), we obtain

$$\|G(u) - G_{r_u}(u)\|_{L^2(\rho_y)} \le 2^{\gamma} L_G r_u^{\gamma} + 2B_{\mathcal{V}} C_U e^{-3M_u r_u^2}.$$

Step 3: Choice of M_u. By the definition of M_u in (32), $M_u = \frac{1}{3r_u^2} \log\left(\frac{2B_{\mathcal{V}} C_U}{L_G r_u^{\gamma}}\right)$. Therefore,

$$2B_{\mathcal{V}} C_U e^{-3M_u r_u^2} = L_G r_u^{\gamma}.$$

Since $u \in \mathcal{U}$ is arbitrary, taking the supremum over u gives

$$\sup_{u \in \mathcal{U}} \|G(u) - G_{r_u}(u)\|_{L^2(\rho_y)} \le (2^{\gamma} + 1) L_G r_u^{\gamma}.$$

This proves (35). □

D.1.2 Proof of Lemma 2

Proof of Lemma 2. Fix an arbitrary $k \in [C_U]$ and $\mathbf{y} \in \Omega_{\mathcal{V}}$.

Step 1: Covering and definition of the output-domain Softmax POU. Since $\{\mathbf{z}_l\}_{l=1}^{C_V}$ is an r_y-cover of $\Omega_{\mathcal{V}}$ under the Euclidean metric, we can choose $l^*(\mathbf{y}) \in \arg\min_{l \in [C_V]} \|\mathbf{y} - \mathbf{z}_l\|_2$ such that $\|\mathbf{y} - \mathbf{z}_{l^*(\mathbf{y})}\|_2 \le r_y$. The output-domain weights are defined in (22) using affine logits. Equivalently, they can be written in the distance-score form

$$\widetilde{\eta}_l(\mathbf{y}) = \frac{\exp\left(M_y \left(r_y^2 - \|\mathbf{y} - \mathbf{z}_l\|_2^2\right)\right)}{\sum_{l'=1}^{C_V} \exp\left(M_y \left(r_y^2 - \|\mathbf{y} - \mathbf{z}_{l'}\|_2^2\right)\right)}, \qquad l \in [C_V]. \tag{66}$$

Then $\widetilde{\eta}_l(\mathbf{y}) \geq 0$ and $\sum_{l=1}^{C_V} \widetilde{\eta}_l(\mathbf{y}) = 1$. To see the equivalence with (22), notice that the common term $r_y^2 - \|\mathbf{y}\|_2^2$ exactly cancels in the Softmax normalization. Indeed,

$$\begin{aligned}\widetilde{\eta}_l(\mathbf{y}) &= \frac{\exp\left(M_y\left(r_y^2 - \|\mathbf{y}\|_2^2\right)\right)\exp\left(2M_y\mathbf{y}^\top\mathbf{z}_l - M_y\|\mathbf{z}_l\|_2^2\right)}{\sum_{l'=1}^{C_V}\exp\left(M_y\left(r_y^2 - \|\mathbf{y}\|_2^2\right)\right)\exp\left(2M_y\mathbf{y}^\top\mathbf{z}_{l'} - M_y\|\mathbf{z}_{l'}\|_2^2\right)} \\ &= \frac{\exp\left(2M_y\mathbf{y}^\top\mathbf{z}_l - M_y\|\mathbf{z}_l\|_2^2\right)}{\sum_{l'=1}^{C_V}\exp\left(2M_y\mathbf{y}^\top\mathbf{z}_{l'} - M_y\|\mathbf{z}_{l'}\|_2^2\right)}.\end{aligned}$$

Step 2: Near and far error estimates. Using the partition-of-unity property, for every $k \in [C_U]$ and $\mathbf{y} \in \Omega_{\mathcal{V}}$,

$$G(u_k)(\mathbf{y}) - \widetilde{v}_k(\mathbf{y}) = G(u_k)(\mathbf{y}) - \sum_{l=1}^{C_V} G(u_k)(\mathbf{z}_l)\widetilde{\eta}_l(\mathbf{y}) = \sum_{l=1}^{C_V}\widetilde{\eta}_l(\mathbf{y})\big(G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)\big).$$

Taking the absolute value and applying the triangle inequality, we obtain

$$\begin{aligned}|G(u_k)(\mathbf{y}) - \widetilde{v}_k(\mathbf{y})| &\leq \sum_{l=1}^{C_V}\widetilde{\eta}_l(\mathbf{y})|G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)| \\ &= \underbrace{\sum_{l:\,\|\mathbf{y}-\mathbf{z}_l\|_2 \leq 2r_y}\widetilde{\eta}_l(\mathbf{y})|G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)|}_{(I)} \\ &\quad + \underbrace{\sum_{l:\,\|\mathbf{y}-\mathbf{z}_l\|_2 > 2r_y}\widetilde{\eta}_l(\mathbf{y})|G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)|}_{(II)}.\end{aligned}$$

For the near-center term (I), Assumption 3 yields

$$|G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)| \leq B_{\mathcal{V}}\|\mathbf{y} - \mathbf{z}_l\|_2^\beta \leq B_{\mathcal{V}}(2r_y)^\beta$$

for all l satisfying $\|\mathbf{y} - \mathbf{z}_l\|_2 \leq 2r_y$. Therefore,

$$(I) \leq 2^\beta B_{\mathcal{V}} r_y^\beta.$$

For the far-center term (II), Assumption 3 gives

$$|G(u_k)(\mathbf{y}) - G(u_k)(\mathbf{z}_l)| \leq |G(u_k)(\mathbf{y})| + |G(u_k)(\mathbf{z}_l)| \leq 2B_{\mathcal{V}}.$$

It remains to bound the total Softmax mass assigned to far centers. By the distance-score representation in (66), the denominator satisfies

$$\sum_{l'=1}^{C_V}\exp\left(M_y\left(r_y^2 - \|\mathbf{y} - \mathbf{z}_{l'}\|_2^2\right)\right) \geq \exp\left(M_y\left(r_y^2 - \|\mathbf{y} - \mathbf{z}_{l^*(\mathbf{y})}\|_2^2\right)\right) \geq 1.$$

If $\|\mathbf{y} - \mathbf{z}_l\|_2 > 2r_y$, then

$$r_y^2 - \|\mathbf{y} - \mathbf{z}_l\|_2^2 < r_y^2 - 4r_y^2 = -3r_y^2.$$

So, $\widetilde{\eta}_l(\mathbf{y}) \leq e^{-3M_y r_y^2}$. Then,

$$\sum_{l:\,\|\mathbf{y}-\mathbf{z}_l\|_2 > 2r_y}\widetilde{\eta}_l(\mathbf{y}) \leq C_V e^{-3M_y r_y^2},$$

Therefore,

$$(II) \leq 2B_{\mathcal{V}}C_V e^{-3M_y r_y^2}.$$

Combining the estimates for (I) and (II), we obtain

$$|G(u_k)(\mathbf{y}) - \widetilde{v}_k(\mathbf{y})| \leq 2^\beta B_{\mathcal{V}} r_y^\beta + 2B_{\mathcal{V}}C_V e^{-3M_y r_y^2}.$$

Step 3: Choice of M_y. By the definition of M_y in (36), $M_y = \frac{1}{3r_y^2}\log\left(\frac{2C_V}{r_y^\beta}\right)$. Therefore,

$$2B_{\mathcal{V}}C_V e^{-3M_y r_y^2} = B_{\mathcal{V}} r_y^\beta.$$

Since $k \in [C_U]$ and $\mathbf{y} \in \Omega_{\mathcal{V}}$ are arbitrary, taking the supremum over k and $\mathbf{y}$ gives

$$\sup_{k\in[C_U]} \|G(u_k) - \widetilde{v}_k\|_{L^\infty(\Omega_{\mathcal{V}})} \le (2^\beta + 1)B_{\mathcal{V}} r_y^\beta.$$

This proves (38). □

D.1.3 Proof of Lemma 3

Proof of Lemma 3. With $\alpha = m_\alpha + \nu$, s_{quad}, and q chosen as in Lemma 3, we have $2s_{\text{quad}} - 1 \ge m_\alpha$. Partitioning $\Omega_{\mathcal{U}} = [0,1]^{d_1}$ into q^{d_1} congruent subcubes and applying the same quadrature rule on each subcube gives a multidimensional compound rule (Davis & Rabinowitz, 1984, Section 5.8).

On each subcube, we use the tensor-product s_{quad}-point Gauss–Legendre rule. The one-dimensional s_{quad}-point Gauss–Legendre rule has positive weights and is exact for polynomials of degree at most $2s_{\text{quad}} - 1$ (Davis & Rabinowitz, 1984, Section 2.7). By the Cartesian product construction (Davis & Rabinowitz, 1984, Section 5.6), the resulting tensor-product rule has $s_{\text{quad}}^{d_1}$ nodes on each subcube and is exact for every polynomial whose degree in each coordinate is at most $2s_{\text{quad}} - 1$.

For each cube K, let p_K be the Taylor polynomial of h of degree m_α about the center of K. The Hölder Taylor remainder gives

$$\|h - p_K\|_{L^\infty(K)} \le C_T q^{-\alpha} \|h\|_{C^\alpha(\Omega_{\mathcal{U}})},$$

where $C_T = C_T(d_1, \alpha) > 0$. Since $2s_{\text{quad}} - 1 \ge m_\alpha$, the quadrature rule is exact on p_K. Moreover, its weights are positive and sum to $|K|$. Hence

$$\begin{aligned}\left|\int_K h(\mathbf{x})\,d\mathbf{x} - Q_K(h)\right| &\le \left|\int_K (h - p_K)(\mathbf{x})\,d\mathbf{x}\right| + |Q_K(h - p_K)| \\ &\le 2|K|\|h - p_K\|_{L^\infty(K)} \\ &\le 2C_T|K|q^{-\alpha}\|h\|_{C^\alpha(\Omega_{\mathcal{U}})}.\end{aligned}$$

Summing over all q^{d_1} cubes and using $\sum_K |K| = 1$ gives

$$\left|\int_{\Omega_{\mathcal{U}}} h(\mathbf{x})\,d\mathbf{x} - Q_q(h)\right| \le 2C_T q^{-\alpha}\|h\|_{C^\alpha(\Omega_{\mathcal{U}})}.$$

The composite tensor-product rule uses $N_q = (s_{\text{quad}} q)^{d_1}$ quadrature points. By the choice of q in Lemma 3, $N_q \le n_x$ and $q \ge \frac{n_x^{1/d_1}}{2s_{\text{quad}}}$. If $N_q < n_x$, append distinct grid points with zero weights. Thus,

$$\left|\int_{\Omega_{\mathcal{U}}} h(\mathbf{x})\,d\mathbf{x} - \sum_{j=1}^{n_x} w_j h(\mathbf{x}_j)\right| \le 2C_T(2s_{\text{quad}})^\alpha \|h\|_{C^\alpha(\Omega_{\mathcal{U}})} n_x^{-\alpha/d_1}.$$

Therefore, the desired estimate holds with $C_{\text{quad},0} := 2C_T(2s_{\text{quad}})^\alpha$.

The constructed weights are nonnegative. Since the quadrature rule is exact for the constant polynomial 1,

$$\sum_{j=1}^{n_x} w_j = \int_{\Omega_{\mathcal{U}}} 1\,d\mathbf{x} = 1,$$

which proves (40).

Since the cover is internal, $u_k \in \mathcal{U}$. By Assumption 2 and the standard product estimate in Hölder spaces,

$$\|uu_k\|_{C^\alpha(\Omega_{\mathcal{U}})} \le C_{\alpha,d_1}\|u\|_{C^\alpha(\Omega_{\mathcal{U}})}\|u_k\|_{C^\alpha(\Omega_{\mathcal{U}})} \le C_{\alpha,d_1} B_{\mathcal{U}}^2.$$

Applying the preceding quadrature estimate to $h = uu_k$ gives

$$\left|\langle u, u_k\rangle_{L^2(\Omega_{\mathcal{U}})} - \langle u, u_k\rangle_{n_x}\right| \le C_{\text{quad},0} C_{\alpha,d_1} B_{\mathcal{U}}^2 n_x^{-\alpha/d_1}.$$

Define

$$C_{\text{quad}} := C_{\text{quad},0} C_{\alpha,d_1} B_{\mathcal{U}}^2. \tag{67}$$

Taking the supremum over $u \in \mathcal{U}$ and the maximum over $k \in [C_U]$ then proves (41). □

D.1.4 PROOF OF LEMMA 5

Proof of Lemma 5. Fix arbitrary $u \in \mathcal{U}$.

Step 1: Error decomposition. By the definitions of G_{r_u} in (34), $\widetilde{G}_{r_u,r_y}$ in (39), and $\widetilde{G}_{r_u,r_y,n_x}$ in (23), the triangle inequality in $L^2(\rho_y)$ gives

$$\begin{aligned}\left\|G(u) - \widetilde{G}_{r_u,r_y,n_x}(u)\right\|_{L^2(\rho_y)} &\le \|G(u) - G_{r_u}(u)\|_{L^2(\rho_y)} \\ &\quad + \left\|G_{r_u}(u) - \widetilde{G}_{r_u,r_y}(u)\right\|_{L^2(\rho_y)} \\ &\quad + \left\|\widetilde{G}_{r_u,r_y}(u) - \widetilde{G}_{r_u,r_y,n_x}(u)\right\|_{L^2(\rho_y)}.\end{aligned}$$

Step 2: Bounds for the three terms. By Lemma 1,

$$\|G(u) - G_{r_u}(u)\|_{L^2(\rho_y)} \le (2^\gamma + 1) L_G r_u^\gamma.$$

For the second term, using (34), (37), and (39),

$$G_{r_u}(u) - \widetilde{G}_{r_u,r_y}(u) = \sum_{k=1}^{C_U} \widetilde{\beta}_k(u)\big(G(u_k) - \widetilde{v}_k\big).$$

Therefore, by the triangle inequality in $L^2(\rho_y)$,

$$\left\|G_{r_u}(u) - \widetilde{G}_{r_u,r_y}(u)\right\|_{L^2(\rho_y)} \le \sum_{k=1}^{C_U} \widetilde{\beta}_k(u)\,\|G(u_k) - \widetilde{v}_k\|_{L^2(\rho_y)} \le \sum_{k=1}^{C_U} \widetilde{\beta}_k(u)\,\|G(u_k) - \widetilde{v}_k\|_{L^\infty(\Omega_\mathcal{V})}.$$

By Lemma 2 and $\sum_{k=1}^{C_U} \widetilde{\beta}_k(u) = 1$,

$$\left\|G_{r_u}(u) - \widetilde{G}_{r_u,r_y}(u)\right\|_{L^2(\rho_y)} \le (2^\beta + 1) B_\mathcal{V} r_y^\beta.$$

It remains to bound the third term. Define the continuous and discrete logits by

$$\ell_k(u) := 2M_u \langle u, u_k\rangle_{L^2(\Omega_\mathcal{U})} - M_u \|u_k\|^2_{L^2(\Omega_\mathcal{U})}, \qquad \ell_{k,n_x}(u) := 2M_u \langle u, u_k\rangle_{n_x} - M_u \|u_k\|^2_{L^2(\Omega_\mathcal{U})}.$$

By Lemma 3,

$$\left|\langle u, u_k\rangle_{L^2(\Omega_\mathcal{U})} - \langle u, u_k\rangle_{n_x}\right| \le C_{\mathrm{quad}} n_x^{-\alpha/d_1}$$

uniformly over $u \in \mathcal{U}$ and $k \in [C_U]$. Therefore, for every $k \in [C_U]$,

$$|\ell_k(u) - \ell_{k,n_x}(u)| = 2M_u \left|\langle u, u_k\rangle_{L^2(\Omega_\mathcal{U})} - \langle u, u_k\rangle_{n_x}\right| \le 2M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}.$$

Define $\mathbf{a}(u) := (\ell_1(u), \dots, \ell_{C_U}(u))$ and $\mathbf{b}(u) := (\ell_{1,n_x}(u), \dots, \ell_{C_U,n_x}(u))$. Then

$$\|\mathbf{a}(u) - \mathbf{b}(u)\|_\infty \le 2M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}.$$

By Lemma 4,

$$\sum_{k=1}^{C_U} |\widetilde{\beta}_k(u) - \widetilde{\beta}_{k,n_x}(u)| = \|\operatorname{softmax}(\mathbf{a}(u)) - \operatorname{softmax}(\mathbf{b}(u))\|_1 \le 2\|\mathbf{a}(u) - \mathbf{b}(u)\|_\infty \le 4M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}.$$

Finally, using (39) and (23),

$$|\widetilde{G}_{r_u,r_y}(u)(\mathbf{y}) - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y})| = \left|\sum_{k=1}^{C_U} \big(\widetilde{\beta}_k(u) - \widetilde{\beta}_{k,n_x}(u)\big)\widetilde{v}_k(\mathbf{y})\right| \le \sum_{k=1}^{C_U} |\widetilde{\beta}_k(u) - \widetilde{\beta}_{k,n_x}(u)|\,|\widetilde{v}_k(\mathbf{y})|.$$

Since $\widetilde{v}_k(\mathbf{y})$ is a convex combination of $\{G(u_k)(\mathbf{z}_l)\}_{l=1}^{C_V}$, Assumption 3 gives $|\widetilde{v}_k(\mathbf{y})| \le B_\mathcal{V}$. Therefore,

$$|\widetilde{G}_{r_u,r_y}(u)(\mathbf{y}) - \widetilde{G}_{r_u,r_y,n_x}(u)(\mathbf{y})| \le 4B_\mathcal{V} M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}.$$

Since ρ_y is a probability distribution, this uniform bound implies

$$\left\|\widetilde{G}_{r_u,r_y}(u) - \widetilde{G}_{r_u,r_y,n_x}(u)\right\|_{L^2(\rho_y)} \le 4B_\mathcal{V} M_u C_{\mathrm{quad}} n_x^{-\alpha/d_1}.$$

Combining the three bounds and taking the supremum over $u \in \mathcal{U}$ proves (42). □

D.2 Proofs for the Transformer Realization

D.2.1 Proof of Lemma 6

Proof of Lemma 6. Let

$$\mathbf{X}(u,\mathbf{y}) := \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{n_x} & \mathbf{0}_{d_1} & \mathbf{0}_{d_1\times(P-n_x-1)} \\ \mathbf{0}_{d_2} & \cdots & \mathbf{0}_{d_2} & \mathbf{y} & \mathbf{0}_{d_2\times(P-n_x-1)} \\ u(\mathbf{x}_1) & \cdots & u(\mathbf{x}_{n_x}) & 0 & \mathbf{0}_{1\times(P-n_x-1)} \end{bmatrix} \in \mathbb{R}^{(d_1+d_2+1)\times P}.$$

Thus, the first n_x columns contain the discretized input information, the (n_x+1)-st column contains the query point $\mathbf{y}$, and all remaining input coordinates are zero.

Choose the shared token-wise affine embedding

$$\mathbf{W}_E = \begin{bmatrix} I_{d_1+d_2+1} \\ 0_{(D-d_1-d_2-1)\times(d_1+d_2+1)} \end{bmatrix} \in \mathbb{R}^{D\times(d_1+d_2+1)}, \qquad \mathbf{b}_E = \mathbf{0}_D.$$

Then

$$\mathbf{S} := \mathbf{W}_E\mathbf{X}(u,\mathbf{y}) + \mathbf{b}_E\mathbf{1}_P^\top \in \mathbb{R}^{D\times P}$$

places the discretized input and query information in the first $d_1 + d_2 + 1$ coordinates and sets all remaining coordinates to zero.

Define $\mathbf{s}_\theta := \big(\sin(\theta_1),\dots,\sin(\theta_P)\big)$ and $\mathbf{c}_\theta := \big(\cos(\theta_1),\dots,\cos(\theta_P)\big)$. Let the fixed structural-positional encoding matrix be

$$\mathbf{P}_{\text{pos}} := \begin{bmatrix} 0_{(d_1+d_2+1)\times P} \\ \mathbf{e}_P^\top \\ 0_{(n_x+3)\times P} \\ \mathbf{1}_P^\top \\ \mathbf{1}_P^\top - \mathbf{e}_P^\top \\ \mathbf{s}_\theta \\ \mathbf{c}_\theta \end{bmatrix} \in \mathbb{R}^{D\times P},$$

where $\mathbf{e}_P \in \mathbb{R}^P$ is the P-th standard basis vector. Here, $\mathbf{e}_P^\top = (0,\dots,0,1)$ is the null-column indicator, while $\mathbf{1}_P^\top - \mathbf{e}_P^\top = (1,\dots,1,0)$ is the active-column indicator.

Finally, define the preprocessing operator by

$$\mathbf{Z}_0 := \mathbf{S} + \mathbf{P}_{\text{pos}} = \mathbf{W}_E\mathbf{X}(u,\mathbf{y}) + \mathbf{b}_E\mathbf{1}_P^\top + \mathbf{P}_{\text{pos}}.$$

This gives exactly the representation in (43).

□

D.2.2 Proof of Lemma 7

Proof of Lemma 7. Set $M_s := \frac{M}{c_P}$, $M_\theta := \frac{4M}{c_P}$. Let

$$\chi_t := 1_{\{t<P\}}, \qquad \nu_t := 1_{\{t=P\}}, \qquad t \in [P].$$

Recall that row $D-3$ of $\mathbf{Z}_0$ is the all-one row, row $D-2$ equals $(1,\dots,1,0)$ and hence stores χ_t, while row d_1+d_2+2 is the null-column indicator and hence stores ν_t.

Step 1: Feature extraction heads. In this step, each head fixes one output feature row r and one output column j. The indices t and j' are used for the source and target columns of the attention matrix. For

$$h = (r-1)P + j, \qquad r \in [n_x+3], \qquad j \in [P],$$

define the source column

$$i_r := \begin{cases} r, & r \in [n_x], \\ n_x+1, & r = n_x+1, \\ 1, & r \in \{n_x+2, n_x+3\}. \end{cases}$$

Set $M_0 := M_\theta\left(1 - \frac{c_P}{2}\right) + M_s$. This step uses exactly $(n_x + 3)P$ heads.

For $j < P$, write

$$j = j(k_j, l_j) = (k_j - 1)C_V + l_j.$$

Choose $\mathbf{w}_{r,j}^\top \in \mathbb{R}^{1\times D}$ by

$$\mathbf{w}_{r,j}^\top := \begin{cases} 2M_u w_r u_{k_j}(\mathbf{x}_r)\mathbf{e}_{d_1+d_2+1}^\top, & r \in [n_x],\ j < P, \\ 2M_y \sum_{a=1}^{d_2} (\mathbf{z}_{l_j})_a \mathbf{e}_{d_1+a}^\top - M_y \|\mathbf{z}_{l_j}\|_2^2 \mathbf{e}_{D-2}^\top, & r = n_x + 1,\ j < P, \\ N_{k_j}\mathbf{e}_{D-2}^\top, & r = n_x + 2,\ j < P, \\ A_{k_j,l_j}\mathbf{e}_{D-2}^\top, & r = n_x + 3,\ j < P, \\ \mathbf{0}_{1\times D}, & j = P. \end{cases}$$

Denote the ideal target features by

$$F_{r,j} := \mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,i_r}.$$

Then

$$F_{r,j} = \begin{cases} B_{r,k_j}(u), & r \in [n_x],\ j < P, \\ Q_{l_j}(\mathbf{y}), & r = n_x + 1,\ j < P, \\ N_{k_j}, & r = n_x + 2,\ j < P, \\ A_{k_j,l_j}, & r = n_x + 3,\ j < P, \\ 0, & j = P. \end{cases}$$

Thus $F_{r,j}$ is the ideal target feature that the first MHA layer targets at row r and column j.

For $j < P$,

$$\|\mathbf{w}_{r,j}\|_\infty \le \begin{cases} 2M_u B_{\mathcal{U}}, & r \in [n_x], \\ (2 + d_2)M_y, & r = n_x + 1, \\ M_u B_{\mathcal{U}}^2, & r = n_x + 2, \\ B_{\mathcal{V}}, & r = n_x + 3. \end{cases}$$

Moreover, for every $t \in [P]$,

$$\left|\mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,t}\right| \le \begin{cases} 2M_u B_{\mathcal{U}}^2, & r \in [n_x], \\ 3d_2 M_y, & r = n_x + 1, \\ M_u B_{\mathcal{U}}^2, & r = n_x + 2, \\ B_{\mathcal{V}}, & r = n_x + 3. \end{cases}$$

Hence, by (44),

$$\|\mathbf{w}_{r,j}\|_\infty \le B_{\mathrm{aff}}, \qquad \left|\mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,t}\right| \le B_{\mathrm{aff}},$$

for every $r \in [n_x + 3]$ and $j, t \in [P]$.

We construct the query and key matrices as

$$\mathbf{Q}_1^h = \begin{bmatrix} \mathbf{e}_{D-3}^\top \\ \mathbf{e}_{D-1}^\top \\ \mathbf{e}_D^\top \\ M_s \sin(\theta_{i_r})\mathbf{e}_{D-3}^\top \\ M_s \cos(\theta_{i_r})\mathbf{e}_{D-3}^\top \end{bmatrix} \in \mathbb{R}^{5\times D}, \qquad \mathbf{K}_1^h = \begin{bmatrix} M_0 \mathbf{e}_{d_1+d_2+2}^\top \\ M_\theta \sin(\theta_j)\mathbf{e}_{D-2}^\top \\ M_\theta \cos(\theta_j)\mathbf{e}_{D-2}^\top \\ \mathbf{e}_{D-1}^\top - \sin(\theta_P)\mathbf{e}_{d_1+d_2+2}^\top \\ \mathbf{e}_D^\top - \cos(\theta_P)\mathbf{e}_{d_1+d_2+2}^\top \end{bmatrix} \in \mathbb{R}^{5\times D}.$$

It follows that

$$\mathbf{q}_{j'}^h := (\mathbf{Q}_1^h \mathbf{Z}_0)_{:,j'} = \begin{bmatrix} 1 \\ \sin(\theta_{j'}) \\ \cos(\theta_{j'}) \\ M_s \sin(\theta_{i_r}) \\ M_s \cos(\theta_{i_r}) \end{bmatrix} \in \mathbb{R}^5.$$

And

$$\mathbf{k}_t^h := (\mathbf{K}_1^h \mathbf{Z}_0)_{:,t} = \begin{bmatrix} 0 \\ M_\theta \sin(\theta_j) \\ M_\theta \cos(\theta_j) \\ \sin(\theta_t) \\ \cos(\theta_t) \end{bmatrix} \in \mathbb{R}^5, \quad t < P, \qquad \mathbf{k}_P^h := (\mathbf{K}_1^h \mathbf{Z}_0)_{:,P} = \begin{bmatrix} M_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Thus

$$s_{t,j'}^h = \begin{cases} M_\theta \cos(\theta_{j'} - \theta_j) + M_s \cos(\theta_t - \theta_{i_r}), & t < P, \\ M_0, & t = P. \end{cases}$$

The source-column locating term satisfies

$$M_s \cos(\theta_t - \theta_{i_r}) = M_s, \qquad t = i_r,$$

and

$$M_s \cos(\theta_t - \theta_{i_r}) \le M_s(1 - c_P), \qquad t < P, \quad t \neq i_r.$$

The output-column locating term satisfies

$$M_\theta \cos(\theta_{j'} - \theta_j) = M_\theta, \qquad j' = j,$$

and

$$M_\theta \cos(\theta_{j'} - \theta_j) \le M_\theta(1 - c_P), \qquad j' \neq j.$$

Choose the value matrix as

$$\mathbf{V}_1^h = \begin{bmatrix} \mathbf{w}_{r,j}^\top \\ \mathbf{0}_{1\times D} \end{bmatrix} \in \mathbb{R}^{2\times D}.$$

Therefore,

$$(\mathbf{V}_1^h \mathbf{Z}_0)_{:,t} = \begin{bmatrix} \mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,t} \\ 0 \end{bmatrix},$$

and

$$\left|(\mathbf{V}_1^h \mathbf{Z}_0)_{1,t}\right| \le B_{\text{aff}}.$$

We separate the target and non-target attention columns. For the target column $j' = j$, for every $t < P$ with $t \neq i_r$,

$$s_{i_r,j}^h - s_{t,j}^h = M_s\left(1 - \cos(\theta_t - \theta_{i_r})\right) \ge c_P M_s = M,$$

and

$$s_{i_r,j}^h - s_{P,j}^h = M_\theta + M_s - M_0 = \frac{c_P}{2} M_\theta = 2M.$$

Therefore,

$$\sum_{t \neq i_r} (\mathbf{A}_1^h)_{t,j} \le \sum_{t \neq i_r} e^{s_{t,j}^h - s_{i_r,j}^h} \le P e^{-M}.$$

Hence,

$$\left|\left(\mathbf{V}_1^h \mathbf{Z}_0 \mathbf{A}_1^h\right)_{1,j} - F_{r,j}\right| = \left|\sum_{t \neq i_r} (\mathbf{A}_1^h)_{t,j} \left(\mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,t} - \mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,i_r}\right)\right| \le 2P B_{\text{aff}} e^{-M}.$$

For a non-target column $j' \neq j$, for every $t < P$,

$$s_{P,j'}^h - s_{t,j'}^h \ge M_0 - M_\theta(1 - c_P) - M_s = \frac{c_P}{2} M_\theta = 2M.$$

Therefore,

$$\sum_{t<P} (\mathbf{A}_1^h)_{t,j'} \le \sum_{t<P} e^{s_{t,j'}^h - s_{P,j'}^h} \le P e^{-2M} \le P e^{-M}.$$

Since

$$\mathbf{w}_{r,j}^\top (\mathbf{Z}_0)_{:,P} = 0,$$

we have

$$\left|\left(\mathbf{V}_1^h \mathbf{Z}_0 \mathbf{A}_1^h\right)_{1,j'}\right| \le P B_{\text{aff}} e^{-M}.$$

Step 2: Indicator and positional encoding heads. Let $h_0 := (n_x+3)P$. The head h_0+1 preserves the all-one row and the active-indicator row. We construct the query and key matrices as

$$\mathbf{Q}_1^{h_0+1} = \begin{bmatrix} \mathbf{e}_{D-2}^\top \\ \mathbf{e}_{d_1+d_2+2}^\top \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \end{bmatrix} \in \mathbb{R}^{5\times D}, \qquad \mathbf{K}_1^{h_0+1} = \begin{bmatrix} M\mathbf{e}_{D-2}^\top \\ M\mathbf{e}_{d_1+d_2+2}^\top \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \end{bmatrix} \in \mathbb{R}^{5\times D}.$$

Then

$$s_{t,j'}^{h_0+1} = M\chi_t\chi_{j'} + M\nu_t\nu_{j'}.$$

Define

$$v_t^{\mathrm{ind}} := \chi_t - (P-1)e^{-M}\nu_t.$$

Choose the value matrix

$$\mathbf{V}_1^{h_0+1} = \begin{bmatrix} \mathbf{e}_{D-3}^\top \\ \mathbf{e}_{D-2}^\top - (P-1)e^{-M}\mathbf{e}_{d_1+d_2+2}^\top \end{bmatrix} \in \mathbb{R}^{2\times D}.$$

Therefore,

$$(\mathbf{V}_1^{h_0+1}\mathbf{Z}_0)_{:,t} = \begin{bmatrix} 1 \\ v_t^{\mathrm{ind}} \end{bmatrix}, \qquad t \in [P].$$

For $j' < P$,

$$\sum_{t=1}^{P}(\mathbf{A}_1^{h_0+1})_{t,j'}v_t^{\mathrm{ind}} = \frac{(P-1)e^M - (P-1)e^{-M}}{(P-1)e^M+1} = \frac{(P-1)(e^M-e^{-M})}{(P-1)e^M+1} =: a_M,$$

whereas for $j' = P$,

$$\sum_{t=1}^{P}(\mathbf{A}_1^{h_0+1})_{t,P}v_t^{\mathrm{ind}} = \frac{(P-1) - e^M(P-1)e^{-M}}{(P-1)+e^M} = 0.$$

Hence,

$$\left(\mathbf{V}_1^{h_0+1}\mathbf{Z}_0\mathbf{A}_1^{h_0+1}\right)_{:,j'} = \begin{cases} \begin{bmatrix} 1 \\ a_M \end{bmatrix}, & j' < P, \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & j' = P. \end{cases}$$

Moreover,

$$1 - a_M = \frac{1+(P-1)e^{-M}}{(P-1)e^M+1} \le 2e^{-M}.$$

Since $P \ge 2$ and $Pe^{-M} \le 1/4$, we have $e^{-M} \le 1/8$ and hence

$$a_M \ge 1 - 2e^{-M} \ge \frac{3}{4}, \qquad a_M^{-1} \le 2.$$

Let $M_{\mathrm{id}} := \frac{M}{c_P}$. The head h_0+2 preserves the sinusoidal positional rows. We use

$$\mathbf{Q}_1^{h_0+2} = \begin{bmatrix} \mathbf{0}_{1\times D} \\ \mathbf{e}_{D-1}^\top \\ \mathbf{e}_D^\top \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \end{bmatrix}, \qquad \mathbf{K}_1^{h_0+2} = \begin{bmatrix} \mathbf{0}_{1\times D} \\ M_{\mathrm{id}}\mathbf{e}_{D-1}^\top \\ M_{\mathrm{id}}\mathbf{e}_D^\top \\ \mathbf{0}_{1\times D} \\ \mathbf{0}_{1\times D} \end{bmatrix},$$

where $\mathbf{Q}_1^{h_0+2}, \mathbf{K}_1^{h_0+2} \in \mathbb{R}^{5\times D}$. Then

$$s_{t,j'}^{h_0+2} = M_{\mathrm{id}}\cos(\theta_t - \theta_{j'}),$$

and, for $t \ne j'$,

$$s_{j',j'}^{h_0+2} - s_{t,j'}^{h_0+2} \ge c_P M_{\mathrm{id}} = M.$$

Consequently,

$$\sum_{t \neq j'} (\mathbf{A}_1^{h_0+2})_{t,j'} \leq Pe^{-M}.$$

Choose

$$\mathbf{V}_1^{h_0+2} = \begin{bmatrix} \mathbf{e}_{D-1}^\top \\ \mathbf{e}_D^\top \end{bmatrix} \in \mathbb{R}^{2\times D}.$$

By the shift-invariant Softmax calculation on the uniform periodic grid, (Shi & Liao, 2026, Proof of Lemma 5),

$$\left(\mathbf{V}_1^{h_0+2} \mathbf{Z}_0 \mathbf{A}_1^{h_0+2}\right)_{:,j'} = \lambda_{\mathrm{id}} \begin{bmatrix} \sin(\theta_{j'}) \\ \cos(\theta_{j'}) \end{bmatrix}, \qquad j' \in [P],$$

where

$$\lambda_{\mathrm{id}} := \frac{\sum_{t=1}^P e^{M_{\mathrm{id}} \cos(\theta_t)} \cos(\theta_t)}{\sum_{t=1}^P e^{M_{\mathrm{id}} \cos(\theta_t)}}.$$

Furthermore,

$$0 \leq 1 - \lambda_{\mathrm{id}} \leq 2(P-1)e^{-M}.$$

Since $Pe^{-M} \leq \frac{1}{4}$, $2(P-1)e^{-M} \leq \frac{1}{2}$. Hence

$$\lambda_{\mathrm{id}} \geq \frac{1}{2}, \qquad \lambda_{\mathrm{id}}^{-1} \leq 2.$$

Step 3: Output projection and feature error bounds. For each head $h \in [H^1]$, write

$$\mathbf{C}^h := \mathbf{V}_1^h \mathbf{Z}_0 \mathbf{A}_1^h \in \mathbb{R}^{2\times P}.$$

For $r \in [n_x + 3]$ and $j, q \in [P]$, define

$$G_{r,j,q} := \left(\mathbf{C}^{(r-1)P+j}\right)_{1,q} \in \mathbb{R}.$$

The P heads assigned to feature row r form the block

$$\mathbf{C}^{(r)} := \begin{bmatrix} G_{r,1,1} & G_{r,1,2} & \cdots & G_{r,1,P} \\ 0 & 0 & \cdots & 0 \\ G_{r,2,1} & G_{r,2,2} & \cdots & G_{r,2,P} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{r,P,1} & G_{r,P,2} & \cdots & G_{r,P,P} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2P\times P}.$$

The remaining two heads give

$$\mathbf{C}_{\mathrm{ind}} := \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ a_M & a_M & \cdots & a_M & 0 \end{bmatrix} \in \mathbb{R}^{2\times P},$$

and

$$\mathbf{C}_\theta := \lambda_{\mathrm{id}} \begin{bmatrix} \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_{P-1}) & \sin(\theta_P) \\ \cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_{P-1}) & \cos(\theta_P) \end{bmatrix} \in \mathbb{R}^{2\times P}.$$

Concatenating all head outputs gives

$$\mathbf{C}_1 := \begin{bmatrix} \mathbf{C}^{(1)} \\ \vdots \\ \mathbf{C}^{(n_x+3)} \\ \mathbf{C}_{\mathrm{ind}} \\ \mathbf{C}_\theta \end{bmatrix} \in \mathbb{R}^{2H^1\times P}.$$

Let

$$\mathbf{I}_{\mathrm{sum}} := [1 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 1 \quad 0] \in \mathbb{R}^{1\times 2P},$$

and define

$$\mathbf{R}_{\text{feat}} := \begin{bmatrix} \mathbf{I}_{\text{sum}} & \mathbf{0}_{1\times 2P} & \cdots & \mathbf{0}_{1\times 2P} \\ \mathbf{0}_{1\times 2P} & \mathbf{I}_{\text{sum}} & \cdots & \mathbf{0}_{1\times 2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times 2P} & \mathbf{0}_{1\times 2P} & \cdots & \mathbf{I}_{\text{sum}} \end{bmatrix} \in \mathbb{R}^{(n_x+3)\times 2(n_x+3)P}.$$

Choose

$$\mathbf{W}_1^O := \begin{bmatrix} \mathbf{R}_{\text{feat}} & \mathbf{0}_{(n_x+3)\times 4} \\ \mathbf{0}_{(D-n_x-7)\times 2(n_x+3)P} & \mathbf{0}_{(D-n_x-7)\times 4} \\ \mathbf{0}_{4\times 2(n_x+3)P} & I_4 \end{bmatrix} \in \mathbb{R}^{D\times 2H^1}.$$

For $r \in [n_x+3]$ and $j \in [P]$, define

$$\widehat{F}_{r,j} := \sum_{j'=1}^{P} G_{r,j',j}.$$

Then the output of the first MHA layer,

$$\widehat{\mathbf{Z}}_1 = \mathbf{W}_1^O \mathbf{C}_1,$$

satisfies, for $j < P$,

$$(\widehat{\mathbf{Z}}_1)_{:,j} = \begin{bmatrix} \widehat{F}_{1,j} \\ \vdots \\ \widehat{F}_{n_x+3,j} \\ \mathbf{0}_{D-n_x-7} \\ 1 \\ a_M \\ \lambda_{\text{id}} \sin(\theta_j) \\ \lambda_{\text{id}} \cos(\theta_j) \end{bmatrix}, \qquad (\widehat{\mathbf{Z}}_1)_{:,P} = \begin{bmatrix} \widehat{F}_{1,P} \\ \vdots \\ \widehat{F}_{n_x+3,P} \\ \mathbf{0}_{D-n_x-7} \\ 1 \\ 0 \\ \lambda_{\text{id}} \sin(\theta_P) \\ \lambda_{\text{id}} \cos(\theta_P) \end{bmatrix}.$$

For each feature row $r \in [n_x+3]$ and each column $j \in [P]$, one feature head contributes the target-column error and the remaining $P-1$ heads contribute non-target-column errors. Hence

$$\begin{aligned} |\widehat{F}_{r,j} - F_{r,j}| &= \left| G_{r,j,j} - F_{r,j} + \sum_{j'\neq j} G_{r,j',j} \right| \\ &\le |G_{r,j,j} - F_{r,j}| + \sum_{j'\neq j} |G_{r,j',j}| \\ &\le 2PB_{\text{aff}} e^{-M} + (P-1)PB_{\text{aff}} e^{-M} \\ &\le 2P^2 B_{\text{aff}} e^{-M} = \varepsilon_{\text{feat}}. \end{aligned}$$

For an active column $j = j(k_j, l_j)$, this gives

$$\max\left\{ \max_{i\in[n_x]} \left|\widehat{F}_{i,j} - B_{i,k_j}(u)\right|, \left|\widehat{F}_{n_x+1,j} - Q_{l_j}(\mathbf{y})\right|, \left|\widehat{F}_{n_x+2,j} - N_{k_j}\right|, \left|\widehat{F}_{n_x+3,j} - A_{k_j,l_j}\right| \right\} \le \varepsilon_{\text{feat}}.$$

Since $F_{r,P} = 0$ for every $r \in [n_x+3]$, define $\delta_i^B := \widehat{F}_{i,P}$, $\delta^Q := \widehat{F}_{n_x+1,P}$, $\delta^N := \widehat{F}_{n_x+2,P}$ and $\delta^A := \widehat{F}_{n_x+3,P}$. Then

$$\max\left\{ \max_{i\in[n_x]} |\delta_i^B|, |\delta^Q|, |\delta^N|, |\delta^A| \right\} \le \varepsilon_{\text{feat}}.$$

Step 4: Parameter bound.

We next bound the parameters of the MHA layer. For the feature extraction heads, (44) gives $\|\mathbf{V}_1^h\|_{\max} \le B_{\text{aff}}$. Moreover, since $M_s = \frac{M}{c_P}$ and $M_\theta = \frac{4M}{c_P}$,

$$|M_0| \le M_\theta + M_s = \frac{5M}{c_P}.$$

Thus, the feature-extraction heads satisfy

$$\|\mathbf{Q}_1^h\|_{\max}, \|\mathbf{K}_1^h\|_{\max} \le \max\left\{1, \frac{5M}{c_P}\right\}, \qquad \|\mathbf{V}_1^h\|_{\max} \le B_{\text{aff}}.$$

The two auxiliary heads satisfy the same parameter bound, and $\|\mathbf{W}_1^O\|_{\max} \le 1$. Therefore,

$$M_{A_1} \le \max\left\{1,\, B_{\text{aff}},\, \frac{5M}{c_P}\right\}.$$

□

D.2.3 Proof of Lemma 8

Proof of Lemma 8. Let $\mathbf{e}_r \in \mathbb{R}^D$ denote the r-th standard basis vector. Define

$$\mathbf{r}_\Xi^\top := \sum_{i=1}^{n_x} \mathbf{e}_i^\top + \mathbf{e}_{n_x+1}^\top - \mathbf{e}_{n_x+2}^\top - M_{\text{out}}\mathbf{e}_{D-3}^\top + a_M^{-1} M_{\text{out}} \mathbf{e}_{D-2}^\top.$$

Also define

$$\mathbf{r}_A^\top := \mathbf{e}_{n_x+3}^\top, \qquad \mathbf{r}_1^\top := \mathbf{e}_{D-3}^\top.$$

Let

$$\mathbf{R}_1 := \begin{bmatrix} \mathbf{r}_\Xi^\top \\ \mathbf{r}_A^\top \\ \mathbf{r}_1^\top \\ \mathbf{0}_{(D-3)\times D} \end{bmatrix} \in \mathbb{R}^{D\times D}.$$

For every active column $j = j(k_j, l_j) \in [P-1]$, using (45),

$$\mathbf{r}_\Xi^\top (\widehat{\mathbf{Z}}_1)_{:,j} = \sum_{i=1}^{n_x} \widetilde{B}_{i,k_j,l_j} + \widetilde{Q}_{k_j,l_j} - \widetilde{N}_{k_j,l_j} - M_{\text{out}} + a_M^{-1} M_{\text{out}} a_M = \widehat{\Xi}_j.$$

Moreover,

$$\mathbf{r}_A^\top (\widehat{\mathbf{Z}}_1)_{:,j} = \widetilde{A}_{k_j,l_j} = \widetilde{A}_j, \qquad \mathbf{r}_1^\top (\widehat{\mathbf{Z}}_1)_{:,j} = 1.$$

For the null column,

$$\mathbf{r}_\Xi^\top (\widehat{\mathbf{Z}}_1)_{:,P} = \sum_{i=1}^{n_x} \delta_i^B - \delta^N + \delta^Q - M_{\text{out}} = \widehat{\Xi}_P,$$

since $(\widehat{\mathbf{Z}}_1)_{D-2,P} = 0$. Also,

$$\mathbf{r}_A^\top (\widehat{\mathbf{Z}}_1)_{:,P} = \delta^A, \qquad \mathbf{r}_1^\top (\widehat{\mathbf{Z}}_1)_{:,P} = 1.$$

Hence

$$\mathbf{R}_1 \widehat{\mathbf{Z}}_1 = \begin{bmatrix} \widehat{\Xi}_1 & \cdots & \widehat{\Xi}_{P-1} & \widehat{\Xi}_P \\ \widetilde{A}_1 & \cdots & \widetilde{A}_{P-1} & \delta^A \\ 1 & \cdots & 1 & 1 \\ & \mathbf{0}_{(D-3)\times P} & & \end{bmatrix}.$$

We realize this linear map by a point-wise ReLU FFN of hidden width $d_{\text{ff}}^1 = 2D$. Choose

$$\mathbf{W}_1^1 = \begin{bmatrix} I_D \\ -I_D \end{bmatrix}, \qquad \mathbf{b}_1^1 = \mathbf{0}_{2D},$$

and

$$\mathbf{W}_1^2 = \begin{bmatrix} \mathbf{R}_1 & -\mathbf{R}_1 \end{bmatrix}, \qquad \mathbf{b}_1^2 = \mathbf{0}_D.$$

Using $x = \sigma(x) - \sigma(-x)$ componentwise, for every $\mathbf{z} \in \mathbb{R}^D$,

$$\mathbf{W}_1^2 \sigma\left(\mathbf{W}_1^1 \mathbf{z} + \mathbf{b}_1^1\right) + \mathbf{b}_1^2 = \mathbf{R}_1 \mathbf{z}.$$

Therefore,

$$\mathbf{Z}_1 = F_1(\widehat{\mathbf{Z}}_1) = \mathbf{R}_1\widehat{\mathbf{Z}}_1,$$

which proves (47).

For every active column $j = j(k_j, l_j)$, the feature bounds in Lemma 7 give

$$\begin{aligned}|\widehat{\Xi}_j - \Xi_j| &\le \sum_{i=1}^{n_x} \left|\widetilde{B}_{i,k_j,l_j} - B_{i,k_j}(u)\right| \\ &\quad + \left|\widetilde{N}_{k_j,l_j} - N_{k_j}\right| + \left|\widetilde{Q}_{k_j,l_j} - Q_{l_j}(\mathbf{y})\right| \\ &\le (n_x+2)\varepsilon_{\text{feat}}.\end{aligned}$$

Similarly,

$$|\widetilde{A}_j - A_{k_j,l_j}| \le \varepsilon_{\text{feat}}.$$

This proves (48).

For the null column,

$$\widehat{\Xi}_P \le -M_{\text{out}} + (n_x+2)\varepsilon_{\text{feat}}.$$

Moreover, for every active $j \in [P-1]$,

$$|\widehat{\Xi}_j| \le (n_x+2)\left(B_{\text{aff}} + \varepsilon_{\text{feat}}\right).$$

Therefore,

$$\widehat{\Xi}_P - \max_{j\in[P-1]} \widehat{\Xi}_j \le -M_{\text{out}} + (n_x+2)\varepsilon_{\text{feat}} + (n_x+2)\left(B_{\text{aff}} + \varepsilon_{\text{feat}}\right) = -M,$$

by the definition of M_{out}. Together with $|\delta^A| \le \varepsilon_{\text{feat}}$ from Lemma 7, this proves (49).

Finally, under $Pe^{-M} \le 1/4$, Lemma 7 gives $a_M^{-1} \le 2$. Since $M_{\text{out}} > M > 1$,

$$\|\mathbf{R}_1\|_{\max} \le 2M_{\text{out}}.$$

Hence the displayed FFN matrices satisfy

$$M_{F_1} \le 2M_{\text{out}},$$

which proves (50).

□

D.2.4 PROOF OF LEMMA 9

Proof of Lemma 9. Choose

$$\mathbf{Q}_2 = \mathbf{e}_3^\top, \qquad \mathbf{K}_2 = \mathbf{e}_1^\top, \qquad \mathbf{V}_2 = \mathbf{e}_2^\top.$$

By the form of $\mathbf{Z}_1$ in (47), for every target column $j' \in [P]$,

$$q_{j'} := (\mathbf{Q}_2\mathbf{Z}_1)_{j'} = 1,$$

while for every source column $j \in [P]$,

$$k_j := (\mathbf{K}_2\mathbf{Z}_1)_j = \widehat{\Xi}_j.$$

Hence the attention scores are

$$s_{j,j'} = k_j q_{j'} = \widehat{\Xi}_j,$$

independent of j', and therefore every target column receives the same attention weights

$$\gamma_j = \frac{\exp(\widehat{\Xi}_j)}{\sum_{q=1}^{P}\exp(\widehat{\Xi}_q)}.$$

By (49),

$$\widehat{\Xi}_P - \max_{j<P} \widehat{\Xi}_j \le -M,$$

and thus

$$\gamma_P \le \exp\left(\widehat{\Xi}_P - \max_{j<P} \widehat{\Xi}_j\right) \le e^{-M}.$$

The value projection satisfies

$$(\mathbf{V}_2\mathbf{Z}_1)_j = \widetilde{A}_j, \qquad j < P,$$

and

$$(\mathbf{V}_2\mathbf{Z}_1)_P = \delta^A.$$

Therefore,

$$\widehat{g}(u,\mathbf{y}) = \sum_{j=1}^{P-1} \gamma_j \widetilde{A}_j + \gamma_P \delta^A.$$

Choosing $\mathbf{W}_2^O = \mathbf{e}_1$ yields $\widehat{\mathbf{Z}}_2 = A_2(\mathbf{Z}_1)$, with the form in (51).

For $j < P$,

$$\gamma_j = (1-\gamma_P)\widehat{\omega}_j,$$

so

$$\sum_{j=1}^{P-1} |\gamma_j - \widehat{\omega}_j| = \gamma_P \le e^{-M}.$$

Using $|\widetilde{A}_j| \le B_{\mathcal{V}} + \varepsilon_{\text{feat}}$ and $|\delta^A| \le \varepsilon_{\text{feat}}$, we obtain

$$\left|\widehat{g}(u,\mathbf{y}) - \sum_{j=1}^{P-1} \widehat{\omega}_j \widetilde{A}_j\right| \le (B_{\mathcal{V}} + 2\varepsilon_{\text{feat}})\, e^{-M}.$$

Finally,

$$\|\mathbf{Q}_2\|_{\max} = \|\mathbf{K}_2\|_{\max} = \|\mathbf{V}_2\|_{\max} = \|\mathbf{W}_2^O\|_{\max} = 1,$$

and hence $M_{A_2} \le 1$. □

D.3 Proof of the Covering Number Bound

Proof of Lemma 10. Let T_Θ and $T_{\widetilde{\Theta}}$ be two Transformer networks in $\mathcal{T}_\epsilon$ parameterized by Θ and $\widetilde{\Theta}$, respectively, where $\|\Theta\|_\infty$, $\|\widetilde{\Theta}\|_\infty \le M_{\max}$. Assume the parameter difference is bounded by $\|\Theta - \widetilde{\Theta}\|_\infty \le \delta$. We aim to bound $\|T_\Theta - T_{\widetilde{\Theta}}\|_{\infty,\infty}$ with respect to δ.

Step 1: Pre-processing and First Encoder Block. By Lemma 6, the fixed preprocessing map gives the same initial representation for the two networks, $\mathbf{Z}_0 = \widetilde{\mathbf{Z}}_0$. Since $\mathbf{x}_i \in [0,1]^{d_1}$, $\mathbf{y} \in [0,1]^{d_2}$, $\|u\|_{L^\infty(\Omega_{\mathcal{U}})} \le B_{\mathcal{U}}$, and all structural-positional entries are bounded by 1, we have

$$r_0 := \|\mathbf{Z}_0\|_{\max} = \|\widetilde{\mathbf{Z}}_0\|_{\max} \le B_{\mathcal{U}}.$$

Let A_1 and F_1 denote the MHA and FFN mappings in the first encoder block. We first consider the concatenated output of the H^1 attention heads before the output projection, denoted by

$$\mathbf{H}_1(\mathbf{Z}_0) \in \mathbb{R}^{(H^1 d_v^1)\times P}.$$

For a single attention head $h \in [H^1]$, define the pre-Softmax score matrix

$$\mathbf{S}_1^h := \mathbf{Z}_0^\top (\mathbf{K}_1^h)^\top \mathbf{Q}_1^h \mathbf{Z}_0 \in \mathbb{R}^{P\times P},$$

the corresponding attention matrix

$$\mathbf{A}_1^h := \mathrm{Softmax}(\mathbf{S}_1^h) \in \mathbb{R}^{P\times P},$$

and the head output

$$\mathbf{H}_1^h(\mathbf{Z}_0) := \mathbf{V}_1^h \mathbf{Z}_0 \mathbf{A}_1^h \in \mathbb{R}^{d_v^1 \times P}.$$

Define $\widetilde{\mathbf{S}}_1^h$, $\widetilde{\mathbf{A}}_1^h$, and $\widetilde{\mathbf{H}}_1^h$ analogously. Recall that

$$d_k^1 = 5, \qquad d_v^1 = 2, \qquad H^1 = (n_x + 3)P + 2.$$

We decompose the difference into a parameter error and an input error:

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

Since the preprocessing map is fixed, $\mathbf{Z}_0 = \widetilde{\mathbf{Z}}_0$, and hence $(\mathrm{II}) = 0$. It remains to bound the parameter error (I).

For the parameter error, fix a head $h \in [H^1]$. Since the input $\mathbf{Z}_0$ is the same for the two parameter choices,

$$\begin{aligned}\left\|(\mathbf{K}_1^h)^\top \mathbf{Q}_1^h - (\widetilde{\mathbf{K}}_1^h)^\top \widetilde{\mathbf{Q}}_1^h\right\|_{\max} &\le d_k^1 \|\mathbf{K}_1^h\|_{\max} \|\mathbf{Q}_1^h - \widetilde{\mathbf{Q}}_1^h\|_{\max} \\ &\quad + d_k^1 \|\mathbf{K}_1^h - \widetilde{\mathbf{K}}_1^h\|_{\max} \|\widetilde{\mathbf{Q}}_1^h\|_{\max} \\ &\le 2 d_k^1 M_{\max} \delta \\ &\le 10 M_{\max} \delta.\end{aligned}$$

Therefore,

$$\begin{aligned}\|\mathbf{S}_1^h - \widetilde{\mathbf{S}}_1^h\|_{\max} &= \left\|\mathbf{Z}_0^\top \left[(\mathbf{K}_1^h)^\top \mathbf{Q}_1^h - (\widetilde{\mathbf{K}}_1^h)^\top \widetilde{\mathbf{Q}}_1^h\right] \mathbf{Z}_0\right\|_{\max} \\ &\le 10 D^2 r_0^2 M_{\max} \delta.\end{aligned}$$

By Lemma 4, for every column $j \in [P]$,

$$\left\|(\mathbf{A}_1^h)_{:,j} - (\widetilde{\mathbf{A}}_1^h)_{:,j}\right\|_1 \le 2\|\mathbf{S}_1^h - \widetilde{\mathbf{S}}_1^h\|_{\max} \le 20 D^2 r_0^2 M_{\max} \delta.$$

Since each attention matrix is column-stochastic,

$$\|(\mathbf{A}_1^h)_{:,j}\|_1 = \|(\widetilde{\mathbf{A}}_1^h)_{:,j}\|_1 = 1.$$

Hence

$$\begin{aligned}\left\|\mathbf{H}_1^h(\mathbf{Z}_0) - \widetilde{\mathbf{H}}_1^h(\mathbf{Z}_0)\right\|_{\max} &= \left\|\mathbf{V}_1^h \mathbf{Z}_0 \mathbf{A}_1^h - \widetilde{\mathbf{V}}_1^h \mathbf{Z}_0 \widetilde{\mathbf{A}}_1^h\right\|_{\max} \\ &\le \left\|(\mathbf{V}_1^h - \widetilde{\mathbf{V}}_1^h)\mathbf{Z}_0 \mathbf{A}_1^h\right\|_{\max} \\ &\quad + \left\|\widetilde{\mathbf{V}}_1^h \mathbf{Z}_0 (\mathbf{A}_1^h - \widetilde{\mathbf{A}}_1^h)\right\|_{\max} \\ &\le D\delta r_0 + D M_{\max} r_0 \max_{j \in [P]} \left\|(\mathbf{A}_1^h)_{:,j} - (\widetilde{\mathbf{A}}_1^h)_{:,j}\right\|_1 \\ &\le \left(D r_0 + 20 D^3 M_{\max}^2 r_0^3\right) \delta \\ &\le 21 D^3 M_{\max}^2 r_0^3 \delta.\end{aligned}$$

Since $\mathbf{H}_1$ is obtained by concatenating the H^1 head outputs, the max norm does not increase under concatenation. Therefore,

$$\left\|\mathbf{H}_1(\mathbf{Z}_0) - \widetilde{\mathbf{H}}_1(\widetilde{\mathbf{Z}}_0)\right\|_{\max} \le 21 D^3 M_{\max}^2 B_{\mathcal{U}}^3 \delta.$$

Moreover,

$$\|\mathbf{H}_1(\mathbf{Z}_0)\|_{\max},\ \|\widetilde{\mathbf{H}}_1(\widetilde{\mathbf{Z}}_0)\|_{\max} \le D M_{\max} B_{\mathcal{U}}.$$

Since $P \ge n_x + 2$,

$$H^1 d_v^1 = 2\big((n_x + 3)P + 2\big) \le 4P^2.$$

The MHA outputs are

$$A_1(\mathbf{Z}_0) = \mathbf{W}_1^O \mathbf{H}_1(\mathbf{Z}_0), \qquad \widetilde{A}_1(\widetilde{\mathbf{Z}}_0) = \widetilde{\mathbf{W}}_1^O \widetilde{\mathbf{H}}_1(\widetilde{\mathbf{Z}}_0).$$

Therefore,

$$\begin{aligned}\left\|A_1(\mathbf{Z}_0) - \widetilde{A}_1(\widetilde{\mathbf{Z}}_0)\right\|_{\max} &\le 4P^2\left(21D^3M_{\max}^3B_{\mathcal{U}}^3 + DM_{\max}B_{\mathcal{U}}\right)\delta \\ &\le 88P^2D^3M_{\max}^3B_{\mathcal{U}}^3\delta.\end{aligned}$$

Moreover,

$$\|A_1(\mathbf{Z}_0)\|_{\max},\ \|\widetilde{A}_1(\widetilde{\mathbf{Z}}_0)\|_{\max} \le 4P^2DM_{\max}^2B_{\mathcal{U}}.$$

For the first point-wise FFN, define

$$\mathbf{Y}_1 := \sigma\left(\mathbf{W}_1^1A_1(\mathbf{Z}_0) + \mathbf{b}_1^1\mathbf{1}_P^\top\right),$$

and

$$\widetilde{\mathbf{Y}}_1 := \sigma\left(\widetilde{\mathbf{W}}_1^1\widetilde{A}_1(\widetilde{\mathbf{Z}}_0) + \widetilde{\mathbf{b}}_1^1\mathbf{1}_P^\top\right).$$

Since ReLU is 1-Lipschitz,

$$\begin{aligned}\|\mathbf{Y}_1 - \widetilde{\mathbf{Y}}_1\|_{\max} &\le \left(4P^2D^2M_{\max}^2B_{\mathcal{U}} + 88P^2D^4M_{\max}^4B_{\mathcal{U}}^3 + 1\right)\delta \\ &\le 93P^2D^4M_{\max}^4B_{\mathcal{U}}^3\delta.\end{aligned}$$

The magnitude is bounded by

$$\|\mathbf{Y}_1\|_{\max},\ \|\widetilde{\mathbf{Y}}_1\|_{\max} \le 5P^2D^2M_{\max}^3B_{\mathcal{U}}.$$

Let

$$\mathbf{Z}_1 = F_1(A_1(\mathbf{Z}_0)), \qquad \widetilde{\mathbf{Z}}_1 = \widetilde{F}_1(\widetilde{A}_1(\widetilde{\mathbf{Z}}_0)).$$

Since $d_{\text{ff}}^1 = 2D$,

$$\begin{aligned}\|\mathbf{Z}_1 - \widetilde{\mathbf{Z}}_1\|_{\max} &\le \left(10P^2D^3M_{\max}^3B_{\mathcal{U}} + 186P^2D^5M_{\max}^5B_{\mathcal{U}}^3 + 1\right)\delta \\ &\le 197P^2D^5M_{\max}^5B_{\mathcal{U}}^3\delta.\end{aligned}$$

Finally, set $r_1 := 11P^2D^3M_{\max}^4B_{\mathcal{U}}$. Then

$$\|\mathbf{Z}_1\|_{\max},\ \|\widetilde{\mathbf{Z}}_1\|_{\max} \le r_1.$$

Moreover, since $P, D, M_{\max}, B_{\mathcal{U}} \ge 1$, we have $r_1 \ge 1$.

Step 2: Second Encoder Block. We apply the same decomposition to the second encoder block, now accounting for both the parameter perturbation and the input perturbation $\mathbf{Z}_1 - \widetilde{\mathbf{Z}}_1$. Since

$$H^2 = 1, \qquad d_k^2 = d_v^2 = 1, \qquad d_{\text{ff}}^2 = 2D,$$

the MHA bound simplifies. Adapting the derivation from Step 1 to the second block gives

$$\begin{aligned}\left\|\mathbf{H}_2(\mathbf{Z}_1) - \widetilde{\mathbf{H}}_2(\widetilde{\mathbf{Z}}_1)\right\|_{\max} &\le 5M_{\max}^2D^3r_1^3\delta + 5M_{\max}^3D^3r_1^2\|\mathbf{Z}_1 - \widetilde{\mathbf{Z}}_1\|_{\max} \\ &\le 5M_{\max}^2D^3\left(11P^2D^3M_{\max}^4B_{\mathcal{U}}\right)^3\delta \\ &\quad + 5M_{\max}^3D^3\left(11P^2D^3M_{\max}^4B_{\mathcal{U}}\right)^2 \\ &\qquad \cdot \left(197P^2D^5M_{\max}^5B_{\mathcal{U}}^3\delta\right) \\ &\le 6655P^6D^{12}M_{\max}^{14}B_{\mathcal{U}}^3\delta \\ &\quad + 119185P^6D^{14}M_{\max}^{16}B_{\mathcal{U}}^5\delta \\ &\le 125840P^6D^{14}M_{\max}^{16}B_{\mathcal{U}}^5\delta.\end{aligned}$$

Its magnitude is bounded by

$$\|\mathbf{H}_2(\mathbf{Z}_1)\|_{\max},\ \|\widetilde{\mathbf{H}}_2(\widetilde{\mathbf{Z}}_1)\|_{\max} \le DM_{\max}r_1 \le 11P^2D^4M_{\max}^5B_{\mathcal{U}}.$$

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

$$\begin{aligned}\left\|A_2(\mathbf{Z}_1) - \widetilde{A}_2(\widetilde{\mathbf{Z}}_1)\right\|_{\max} &\le M_{\max}\left\|\mathbf{H}_2(\mathbf{Z}_1) - \widetilde{\mathbf{H}}_2(\widetilde{\mathbf{Z}}_1)\right\|_{\max} \\ &\quad + \delta\|\widetilde{\mathbf{H}}_2(\widetilde{\mathbf{Z}}_1)\|_{\max} \\ &\le M_{\max}\left(125840P^6D^{14}M_{\max}^{16}B_{\mathcal{U}}^5\delta\right) \\ &\quad + \delta\left(11P^2D^4M_{\max}^5 B_{\mathcal{U}}\right) \\ &\le 125851P^6D^{14}M_{\max}^{17}B_{\mathcal{U}}^5\delta.\end{aligned}$$

Moreover,

$$\begin{aligned}\|A_2(\mathbf{Z}_1)\|_{\max},\, \|\widetilde{A}_2(\widetilde{\mathbf{Z}}_1)\|_{\max} &\le M_{\max}\max\left\{\|\mathbf{H}_2(\mathbf{Z}_1)\|_{\max}, \|\widetilde{\mathbf{H}}_2(\widetilde{\mathbf{Z}}_1)\|_{\max}\right\} \\ &\le 11P^2D^4M_{\max}^6 B_{\mathcal{U}}.\end{aligned}$$

For the second point-wise FFN, define

$$\mathbf{Y}_2 := \sigma\left(\mathbf{W}_2^1 A_2(\mathbf{Z}_1) + \mathbf{b}_2^1\mathbf{1}_P^\top\right), \qquad \widetilde{\mathbf{Y}}_2 := \sigma\left(\widetilde{\mathbf{W}}_2^1\widetilde{A}_2(\widetilde{\mathbf{Z}}_1) + \widetilde{\mathbf{b}}_2^1\mathbf{1}_P^\top\right).$$

Their magnitudes satisfy

$$\begin{aligned}\|\mathbf{Y}_2\|_{\max},\, \|\widetilde{\mathbf{Y}}_2\|_{\max} &\le DM_{\max}\max\left\{\|A_2(\mathbf{Z}_1)\|_{\max}, \|\widetilde{A}_2(\widetilde{\mathbf{Z}}_1)\|_{\max}\right\} + M_{\max} \\ &\le 12P^2D^5M_{\max}^7 B_{\mathcal{U}}.\end{aligned}$$

Since ReLU is 1-Lipschitz,

$$\begin{aligned}\|\mathbf{Y}_2 - \widetilde{\mathbf{Y}}_2\|_{\max} &\le D\delta\|\widetilde{A}_2(\widetilde{\mathbf{Z}}_1)\|_{\max} \\ &\quad + DM_{\max}\left\|A_2(\mathbf{Z}_1) - \widetilde{A}_2(\widetilde{\mathbf{Z}}_1)\right\|_{\max} + \delta \\ &\le D\delta\left(11P^2D^4M_{\max}^6 B_{\mathcal{U}}\right) \\ &\quad + DM_{\max}\left(125851P^6D^{14}M_{\max}^{17}B_{\mathcal{U}}^5\delta\right) + \delta \\ &\le 125863P^6D^{15}M_{\max}^{18}B_{\mathcal{U}}^5\delta.\end{aligned}$$

Let

$$\mathbf{Z}_2 = F_2(A_2(\mathbf{Z}_1)), \qquad \widetilde{\mathbf{Z}}_2 = \widetilde{F}_2(\widetilde{A}_2(\widetilde{\mathbf{Z}}_1)).$$

Since $d_{\text{ff}}^2 = 2D$, it follows that

$$\begin{aligned}\|\mathbf{Z}_2 - \widetilde{\mathbf{Z}}_2\|_{\max} &\le 2D\delta\|\widetilde{\mathbf{Y}}_2\|_{\max} + 2DM_{\max}\|\mathbf{Y}_2 - \widetilde{\mathbf{Y}}_2\|_{\max} + \delta \\ &\le 2D\delta\left(12P^2D^5M_{\max}^7 B_{\mathcal{U}}\right) \\ &\quad + 2DM_{\max}\left(125863P^6D^{15}M_{\max}^{18}B_{\mathcal{U}}^5\delta\right) + \delta \\ &\le 251751P^6D^{16}M_{\max}^{19}B_{\mathcal{U}}^5\delta.\end{aligned}$$

Finally, define

$$r_2 := \max\left\{\|\mathbf{Z}_2\|_{\max}, \|\widetilde{\mathbf{Z}}_2\|_{\max}\right\}.$$

Then

$$\begin{aligned}r_2 &\le 2DM_{\max}\max\left\{\|\mathbf{Y}_2\|_{\max}, \|\widetilde{\mathbf{Y}}_2\|_{\max}\right\} + M_{\max} \\ &\le 25P^2D^6M_{\max}^8 B_{\mathcal{U}}.\end{aligned}$$

Step 3: Readout Layer and Covering Number. The final output is computed by a linear readout vector $\mathbf{c} \in \mathbb{R}^{PD}$ with $\|\mathbf{c}\|_\infty \le M_{\max}$. Hence,

$$\begin{aligned}\left|T_\Theta(S_{n_x}(u), \mathbf{y}) - T_{\widetilde{\Theta}}(S_{n_x}(u), \mathbf{y})\right| &\le \|\mathbf{c}\|_1\|\mathbf{Z}_2 - \widetilde{\mathbf{Z}}_2\|_{\max} + \|\mathbf{c} - \widetilde{\mathbf{c}}\|_1\|\widetilde{\mathbf{Z}}_2\|_{\max} \\ &\le (PDM_{\max})\left(251751P^6D^{16}M_{\max}^{19}B_{\mathcal{U}}^5\delta\right) \\ &\quad + (PD\delta)\left(25P^2D^6M_{\max}^8 B_{\mathcal{U}}\right) \\ &\le 251776P^7D^{17}M_{\max}^{20}B_{\mathcal{U}}^5\delta.\end{aligned}$$

Since the preceding bound is uniform over $u \in \mathcal{U}$ and $\mathbf{y} \in \Omega_{\mathcal{Y}}$, we obtain

$$\|T_\Theta - T_{\widetilde{\Theta}}\|_{\infty,\infty} \le L_{\mathrm{par}}\delta,$$

where

$$L_{\mathrm{par}} := 251776P^7D^{17}M_{\max}^{20}B_{\mathcal{U}}^5.$$

To obtain an η-cover of $\mathcal{T}_\epsilon$ in the $\|\cdot\|_{\infty,\infty}$ norm, it suffices to construct a δ_η-cover of the parameter cube $[-M_{\max}, M_{\max}]^{N_{\mathrm{total}}}$, where

$$\delta_\eta := \frac{\eta}{L_{\mathrm{par}}}.$$

Since $\eta \le 1$, $L_{\mathrm{par}} \ge 1$, and $M_{\max} \ge 1$, we have $\delta_\eta \le M_{\max}$. Hence the parameter cube admits an ℓ^∞ δ_η-cover of cardinality at most $\left(\frac{2M_{\max}}{\delta_\eta}\right)^{N_{\mathrm{total}}}$.

Therefore,

$$\log \mathcal{N}\left(\eta, \mathcal{T}_\epsilon, \|\cdot\|_{\infty,\infty}\right) \le N_{\mathrm{total}} \log\left(\frac{2M_{\max}L_{\mathrm{par}}}{\eta}\right) = N_{\mathrm{total}} \log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\eta}\right).$$

Finally, since the clipping map $\Pi_{B_{\mathcal{Y}}}$ is 1-Lipschitz, clipping does not increase the covering number. Hence

$$\log \mathcal{N}\left(\eta, \mathcal{H}_\epsilon^{\mathrm{clip}}, \|\cdot\|_{\infty,\infty}\right) \le N_{\mathrm{total}} \log\left(\frac{503552P^7D^{17}M_{\max}^{21}B_{\mathcal{U}}^5}{\eta}\right).$$

This proves (63). □

E Additional Experimental Details

We provide additional implementation, evaluation, and prediction details for the Burgers' and KdV experiments in Section 5.

E.1 Common Experimental Setup

Both experiments use a two-block encoder-only Transformer with model dimension 192, four attention heads per block, and feedforward dimension 384. The architecture is fixed across sample sizes and intrinsic dimensions within each problem. Table 1 summarizes the training and evaluation settings.

Table 1: Training and evaluation settings for the Burgers' and KdV experiments.

	Burgers'	KdV
Initial learning rate	10^{-3}	3×10^{-4}
Batch size	256	64
Random output queries per training function	8	16
Maximum epochs	12,000	6,000
Test functions	128	128
Test spatial grid	1024 uniform points	1024 uniform points
Repeats per $(d_{\mathcal{U}}, n)$	5	5

All models are trained with Adam. All experiments use noiseless training targets. Learning rates are reduced on validation plateaus, and early stopping and checkpoint selection are based only on validation error. The sample size n denotes the number of independently sampled training functions rather than the number of query–function pairs.

For both equations, the reported test MSE averages squared prediction errors in physical units over 128 held-out functions and 1024 uniformly spaced spatial points. This uniform-grid evaluation provides a discrete estimate of the population risk with ρ_y taken to be uniform over the spatial domain. At each $(d_{\mathcal{U}}, n)$, the plotted marker is the geometric mean of the five test MSEs, the error bar is one sample standard deviation in log MSE, and the exponent p is obtained from the equal-weight log-space fit

$$\log E(n) = a - p \log n. \tag{68}$$

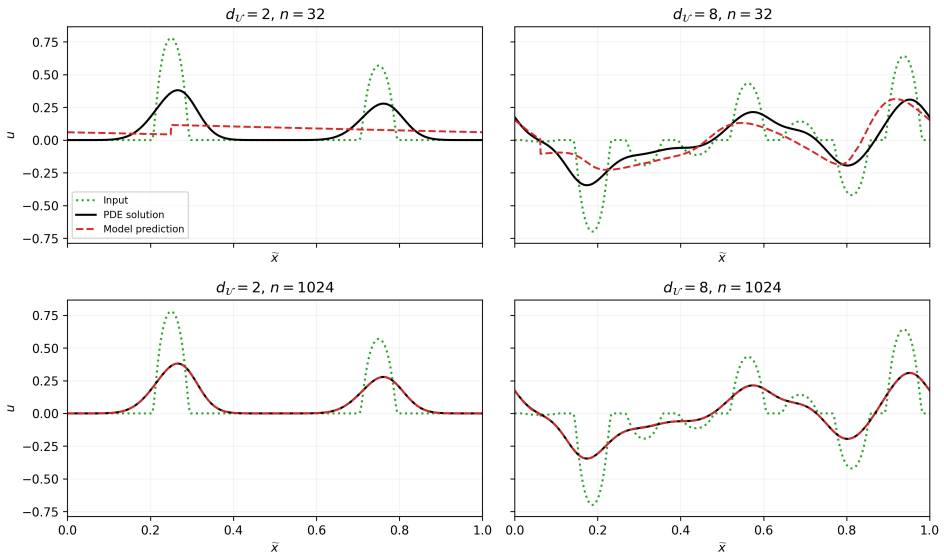

Figure 4: Representative Burgers' predictions. The top and bottom rows correspond to $n = 32$ and $n = 1024$, respectively; the left and right columns correspond to $d_{\mathcal{U}} = 2$ and 8. Dotted curves show inputs, solid curves reference solutions, and dashed curves model predictions. Each column uses the same held-out test input across both sample sizes.

E.2 Viscous Burgers' Equation

For Burgers', we consider

$$u_t + uu_x = \nu u_{xx}, \tag{69}$$

with $\nu = 0.02$ and $T = 0.05$. The input family is

$$u_0(x;\theta) = \sum_{j=0}^{d_{\mathcal{U}}-1} \theta_j \phi_j(x), \qquad \theta_j \overset{\text{i.i.d.}}{\sim} \text{Unif}[-0.25, 0.25]. \tag{70}$$

To define the basis functions, let

$$b_h(r) = \begin{cases} \exp\left[-\dfrac{1}{1-(r/h)^2}\right], & |r| < h, \\ 0, & |r| \geq h, \end{cases} \qquad h = 0.05. \tag{71}$$

We set

$$\phi_j(x) = \frac{b_h(r_j(x))}{Z_h}, \qquad r_j(x) = ((x - j/d_{\mathcal{U}} + 1/2) \bmod 1) - 1/2, \tag{72}$$

for $j = 0, \ldots, d_{\mathcal{U}} - 1$, where Z_h is the periodic L^2 normalization constant, computed on a uniform 65,536-point grid. We consider $d_{\mathcal{U}} \in \{2, 8\}$ and $n \in \{32, 64, 128, 256, 512, 1024\}$. The smaller training sets are nested subsets of the corresponding largest training pool.

The training and validation targets are generated using a Fourier pseudo-spectral method with two-thirds dealiasing and fourth-order Runge–Kutta time stepping. For the uniform-grid test evaluation, we recompute the held-out reference solutions using the periodic Cole–Hopf transformation with a 4096-point Fourier discretization, validated by spatial refinement and comparison with an independent Fourier–Runge–Kutta solver. This reevaluation uses the originally trained checkpoints without retraining. We use a fixed subset of 128 functions from the original held-out test set for every sample size.

Figure 4 uses repeat 0 and fixed test sample 0 within each intrinsic dimension. For visualization, the Burgers' periodic domain is shifted and displayed using the normalized coordinate $\widetilde{x} \in [0, 1]$.

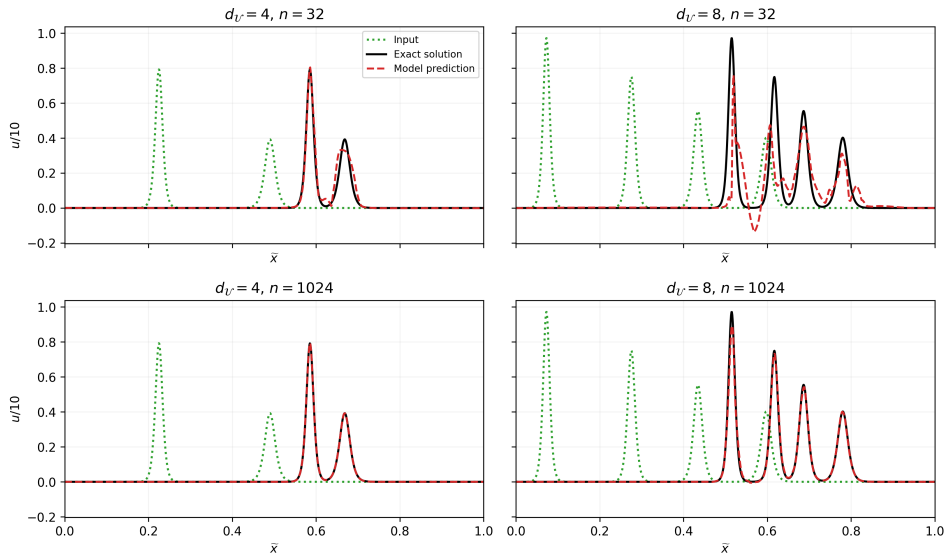

Figure 5: Representative KdV predictions. The top and bottom rows correspond to $n = 32$ and $n = 1024$, respectively; the left and right columns correspond to $d_{\mathcal{U}} = 4$ and 8. Dotted curves show inputs, solid curves exact solutions, and dashed curves model predictions. Each column uses the same held-out test input across both sample sizes.

E.3 KdV Equation

For KdV, we consider

$$u_t + 6uu_x + u_{xxx} = 0 \tag{73}$$

on $x \in [0, 44]$ with $T = 1$. We use exact N-soliton solution families parameterized by the speed and phase of each soliton. Writing

$$\theta = (c_1, p_1, \ldots, c_N, p_N), \tag{74}$$

let $u^{(N)}(x, t; \theta)$ denote the corresponding exact N-soliton solution and set

$$u_0(x; \theta) = u^{(N)}(x, 0; \theta),$$

so that $d_{\mathcal{U}} = 2N$. We take $N = 2$ and 4, giving $d_{\mathcal{U}} = 4$ and 8, respectively. The parameters are sampled independently according to

$$c_j \sim \mathrm{Unif}[0.97\bar{c}_j, 1.03\bar{c}_j], \qquad p_j \sim \mathrm{Unif}[\bar{p}_j - 0.8, \bar{p}_j + 0.8]. \tag{75}$$

For $d_{\mathcal{U}} = 4$, the nominal parameters are $(\bar{c}_j) = (16, 8)$ and $(\bar{p}_j) = (10, 20)$; for $d_{\mathcal{U}} = 8$, they are $(\bar{c}_j) = (20, 15, 11, 8)$ and $(\bar{p}_j) = (4, 10, 16, 22)$. We use $n \in \{32, 64, 128, 256, 512, 1024\}$.

The training, validation, and test targets are generated from the exact multi-soliton solution. During training, both input and output amplitudes are divided by 10; predictions are multiplied by 10 before computing the test MSE in physical units. For visualization, we use $\widetilde{x} = x/44$ and display the amplitude as $u/10$.

Figure 5 uses repeat 0 and fixed test sample 0 within each intrinsic dimension, with validation-selected checkpoints and no visual sample selection.