Journal of Machine Learning Research 27 (2026) 1-34 Submitted 5/25; Revised 9/25; Published 1/26

Transformers Can Overcome the Curse of Dimensionality: A Theoretical Study from an Approximation Perspective

Yuling Jiao YULINGJIAOMATH@WHU.EDU.CN
School of Artificial Intelligence and School of Mathematics,
and National Center for Applied Mathematics in Hubei,
and Hubei Key Laboratory of Computational Science
Wuhan University
Wuhan, 430072, Hubei, China

Yanming Lai* YANMILAI@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Hong Kong, China

Yang Wang YANG.WANG@HKU.HK
Department of Mathematics
The University of Hong Kong
Pokfulam, Hong Kong, China

Bokai Yan BYANAC@CONNECT.UST.HK
Department of Mathematics
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, China

Editor: Mahdi Soltanolkotabi

Abstract

The Transformer model is widely used in various application areas of machine learning, such as natural language processing. This paper investigates the approximation of the Hölder continuous function class $\mathcal{H}_Q^{\beta}\left([0,1]^{d\times n},\mathbb{R}^{d\times n}\right)$ by Transformers and constructs several Transformers that can overcome the curse of dimensionality. These Transformers consist of one self-attention layer with one head and the softmax function as the activation function, along with several feedforward layers. For example, to achieve an approximation accuracy of ϵ, if the activation functions of the feedforward layers in the Transformer are ReLU and floor, only $\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$ layers of feedforward layers are needed, with widths of these layers not exceeding $\mathcal{O}\left(\frac{1}{\epsilon^{2/\beta}}\log\frac{1}{\epsilon}\right)$. If other activation functions are allowed in the feedforward layers, the width of the feedforward layers can be further reduced to a constant. These results demonstrate that Transformers have a strong expressive capability. The construction in this paper is based on the Kolmogorov-Arnold Superposition Theorem and does not require the concept of contextual mapping, hence our proof is more intuitively clear compared to previous Transformer approximation works. Additionally, the translation technique proposed in this paper helps to apply the previous approximation results of feedforward neural networks to Transformer research.

Keywords: Transformer, curse of dimensionality, approximation, self-attention, feedforward neural network

*. Corresponding author

1. Introduction

The Transformer model with the self-attention mechanism, proposed by Vaswani et al. (2017), has been widely applied in deep learning since its inception. Its extensive applications span across multiple fields, including natural language processing (Radford et al., 2018, 2019; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Brown et al., 2020; Fedus et al., 2022), image processing (Dosovitskiy et al., 2021; Ying et al., 2021), generative tasks (Peebles and Xie, 2023), reinforcement learning (Parisotto et al., 2020; Chen et al., 2021; Janner et al., 2021), video understanding (Akbari et al., 2021; Bertasius et al.), and more. The main advantage of Transformers lies in their efficient scalability, which is achieved through parallel token processing, parameter sharing, and token interaction based on simple dot products. Despite its successes in various domains, a comprehensive theoretical understanding of the Transformer model is still evolving.

One of the key reasons for the success of neural network models, including Transformer models, is their ability to broadly represent various functions. The universal approximation theorem is a classic result in neural network theory that dates back several decades (Cybenko, 1989; Hornik, 1991; Funahashi, 1989). These results indicate that as long as the width is large enough, a two-layer feedforward neural network (FNN) can approximate any continuous function with compact support to arbitrary precision. In recent years, there has been significant progress in the theoretical study of approximation for FNNs. From shallow to deep networks, from ReLU to sigmoid activations, various studies have explored the approximation capabilities of different FNNs. The readers are referred to Yarotsky (2017); Lu et al. (2017); Yarotsky (2018); Suzuki (2019); Shen et al. (2020); Lu et al. (2021); Shen et al. (2022); Gühring et al. (2020); Gühring and Raslan (2021); Siegel and Xu (2024) and the references therein. These studies cover a variety of function spaces such as Holder continuous spaces, C^s spaces, Sobolev spaces, Besov spaces, Barron spaces, revealing the broad expressive power of FNNs. Furthermore, some research indicates that by allowing activations other than ReLU and sigmoid in network settings, constructed FNNs can overcome the curse of dimensionality, which is a term proposed by Bellman Bellman (1952) to depict an exponentially increasing difficulty of problems with increasing dimension. These FNNs include such as ReLU-floor networks (Shen et al., 2021a), ReLU-2^x-sine networks (Jiao et al., 2023a), ReLU-sine networks (Yarotsky and Zhevnerchuk, 2020), two types of three-layer networks proposed by (Shen et al., 2021b), where the sizes of all these networks do not grow exponentially with the dimensionality. Interestingly, Yarotsky (2021); Zhang et al. (2022); Wang et al. (2025) have even constructed several networks with their size independent of the approximation accuracy.

Studying the expressive power of Transformers is more challenging compared to FNNs. The main challenge in proving the universal approximation theorem for Transformer models is that Transformers need to consider the context of the entire input sequence. Unlike in FNNs, where each input is processed independently, the self-attention mechanism in Transformer models must consider the dependencies between all tokens in each input sequence. Yun et al. (2020a) introduced the concept of "contextual mapping" to aggregate these dependencies into a token-based quantity through the self-attention mechanism, which is then mapped by a feedforward neural network to the desired output, thus first proving the universal approximation theorem for Transformer models. They demonstrated that if the number of self-attention layers in a Transformer is proportional to the power of the length n of each input sequence, it can approximate continuous functions on compact support regions. This result was later extended to sparse-attention Transformers (Zaheer et al., 2020; Yun et al., 2020b) and Transformers with constraints (Kratsios et al., 2022). Kim et al. (2023) improved the results of Yun et al. (2020a), showing that $2n$ layers of self-attention are sufficient for memorizing finite samples. Based on the concept of contextual mapping and the properties of the Boltzmann operator composed of softmax, Kajitsuka and Sato (2024) demonstrated that a single-layer Transformer with only one head has memorization capabilities, and a Transformer composed of one self-attention layer and two feedforward layers can approximate permutation equivariant functions on compact sets. Hu et al. (2025) extended the results of Kajitsuka and Sato (2024) by generalizing the latter's rank-1 weight matrices to weight matrices of arbitrary rank. Recently, Jiang and Li (2024) explored the approximation capabilities of Transformer networks for Hölder and Sobolev functions, and apply these results to address nonparametric regression estimation with dependent observations.

In addition, Wei et al. (2022) investigated the expressive power of Transformers under statistical learnability constraints. Gurevych et al. (2022) analyzed the theoretical performance of Transformers from the perspective of a hierarchical compositional model. Subsequently, Takakura and Suzuki (2023) extended their results, demonstrating that a one-layer Transformer with an embedding layer is the universal approximator of a shift-equivariant γ smooth function. Petrov et al. (2024) showed that prompting and prefix-tuning a pretrained model can universally approximate sequence-to-sequence functions. Luo et al. (2022) proved that there exist functions that cannot be approximated by Transformers with positional encodings. From the perspective of sequence modeling, Wang and E (2024); Jiang and Li (2024) investigated the expressive power of the Transformer and compared it with that of RNNs. Works on Transformer approximation of sparse functions include Edelman et al. (2022); Bhattamishra et al. (2023); Sanford et al. (2024); Trauger and Tewari (2024); Wang et al. (2024). More research on the memory capabilities of Transformers can be found in Mahdavi et al. (2024); Madden et al. (2024); Chen and Zou (2024); Kajitsuka and Sato (2025).

There are some other works that studied the expressive power of Transformers from different perspectives. Vuckovic et al. (2020) and Kim et al. (2021) investigated the Lipschitz smoothness of attention operations. Bhojanapalli et al. (2020) demonstrated that the small size of attention heads limits the rank of the self-attention matrix. Wies et al. (2021) pinpointed the rank of the input embedding matrix as a bottleneck limiting the impact of network width on expressive capacity. Dong et al. (2021) showed that without skip connections or feedforward layers, the rank of self-attention layers decreases exponentially. Likhosherstov et al. (2023) highlighted that self-attention modules have the ability to replicate any sparse pattern.

In addition to the model's expressive power (approximation error), theoretical analysis of models in machine learning tasks also includes the following two important directions: generalization error, which reflects the model's performance on unseen data, i.e., the performance gap between training data and test data; and optimization error, which is the gap between the model parameters found by the optimization algorithm and the theoretically optimal parameters. In machine learning tasks such as natural language processing and image processing, investigating the performance of Transformers in these aspects can provide a more comprehensive understanding of their successes and limitations.

1.1 Our Contributions

The seminal works on transformer approximation (Yun et al., 2020a; Kim et al., 2023; Kajitsuka and Sato, 2024) focused on the approximation of continuous functions using Transformers, with results all suffering from the curse of dimensionality. A natural question arises: can we construct a Transformer that overcomes the curse of dimensionality under the weakest possible assumptions about the target function? Can the previously established FNNs that have overcome the curse of dimensionality be utilized by us? This paper addresses these questions. Starting from the Kolmogorov–Arnold Representation Theorem, this paper transforms the approximation problem of Transformers into approximation and memorization problems of FNNs, constructing several Transformers that can overcome the curse of dimensionality. This reveals the exceptional expressive power of Transformers

Reference	Use Softmax?	Approximation Rate?	Overcome the Curse of Dimensionality?
Yun et al. (2020a)	✓	✗	✗
Kajitsuka and Sato (2024)	✓	✗	✗
Fang et al. (2022)	✗	✗	✗
Gurevych et al. (2022)	✗	✓	✓
Jiao et al. (2024)	✗	✓	✗
Havrilla and Liao (2024)	✗	✓	✗
Jiao et al. (2025)	✓	✓	✗
Ours (Theorem 6)	✓	✓	✓
Ours (Theorem 7)	✓	✓	✓

Table 1: Comparison of approximation capabilities among various Transformer models.

and partly explains the significant success of Transformers in practical applications. Specifically, the contributions of this paper are as follows:

- This paper constructs several transformers that overcome the curse of dimensionality. To the best of the author's knowledge, this is the first work demonstrating that transformers can overcome the curse of dimensionality. Furthermore, based on the construction presented in this paper, more transformers of this kind can be generated. The constructed transformers require only one layer of self-attention, with both the depth and width of the feedforward layers being small. We only need to assume that the target function is Hölder continuous, which is a very weak assumption. Therefore, the results of this paper have a broad range of applicability.

- Previous works solely focused on approximations within the L^p distance. This paper not only introduces transformers capable of approximating the target function within the L^p distance (Theorem 7), but also develops transformers that can approximate the target function within the L^∞ distance (Theorem 6).

- The construction in this paper is based on the Kolmogorov–Arnold Representation Theorem and does not require the use of the concept of contextual mapping. Therefore, the proof process in this paper is more intuitive and clear compared to previous transformer approximation works. This offers a new approach to studying the approximation properties of transformers.

- How to directly apply the previous results of FNNs to the study of transformers is an important technical issue addressed in this paper. In the conventional setting of the feedforward layer in transformers, the layer applies the same operation to each column of the input matrix. This paper introduces a translation technique that allows the feedforward layer of the transformer to perform different operations on each column of the input matrix, enabling the application of the Kolmogorov–Arnold Representation Theorem. For details, please refer to Section 3.2.

It is worth noting that Jiang and Li (2024); Petrov et al. (2024) also employed the Kolmogorov–Arnold Representation Theorem to investigate the expressive power of Transformers. However, there are several fundamental differences between our work and theirs. The differences between us and Jiang and Li (2024) are as follows. First, the problem formulations differ: we study an approximation problem mapping from $\mathbb{R}^{d\times n}$ to $\mathbb{R}^{d\times n}$, whereas they examine sequence modeling problems involving temporal indices s; Second, the Transformer architectures differ: our definition aligns with classical theoretical works (Yun et al., 2020a; Kajitsuka and Sato, 2024), while they adopt alternative formulations that notably omit specification of activation functions in feedforward layers. The differences between us and Petrov et al. (2024) are as follows. First, their required parameter complexity is $\mathcal{O}(\epsilon^{-10-14m-4m^2})$, with m being the spatial dimension, whereas our maximum parameter complexity is $\mathcal{O}\left(\left(\frac{1}{\epsilon}\right)^{4/\beta}\log^3\frac{1}{\epsilon}\right)$. Therefore, our results successfully overcome the curse of dimensionality while theirs do not. Second, our application of KST differs from theirs, and our proof techniques are entirely independent. We employ translation techniques, while they utilize what they term "core attention head" methodology. Third, their required number of attention layers exhibits linear dependence on input sequence length, while our approach requires only a single attention layer. Given these fundamental differences, our work constitutes an independent contribution.

We compare our work with existing studies in Table 1. A more detailed comparison and a summary of our main results are provided in Table 2.

1.2 Organization of This Paper

The organization of this paper is as follows. In Section 2, we introduce the architecture of our transformer, the Kolmogorov-Arnold Superposition Theorem, and the notations we use. Our main results are presented in Section 3, along with the basic ideas of our construction and formal proofs. In Section 4, we conduct experiments testing the stability of the proposed Transformers during training. In Section 5, we summarize and conclude the paper. The proofs of all auxiliary lemmas can be found in the appendix.

2. Preliminaries

2.1 Transformer

The Transformer model, first introduced by Vaswani et al. (2017), is a mapping which takes a sequence $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ composed of n tokens each with an embedding dimension of size d as an input and outputs another sequence with the same size. In a Transformer model, there are two primary layers: a self-attention layer and a token-wise feedforward layer. The self-attention layer recalibrates each token embedding by aggregating information from all tokens in the sequence, calculated through dot-products of token embeddings. Subsequently, the token-wise feed-forward layer operates independently on these adjusted token embeddings without inter-token communication. Notably, Transformers utilize pairwise dot-products exclusively to capture token interactions within the input sequence.

The Transformer we focus on in this paper is composed of alternating feedforward blocks and self-attention layers. A self-attention layer $\boldsymbol{\mathcal{F}}_{SA} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ with h heads is defined as

$$\boldsymbol{\mathcal{F}}_{SA}(\boldsymbol{X}) := \boldsymbol{X} + \sum_{i=1}^{h} \boldsymbol{W}_O^{(i)}\boldsymbol{W}_V^{(i)}\boldsymbol{X}\sigma_S\left[\left(\boldsymbol{W}_K^{(i)}\boldsymbol{X}\right)^\top\left(\boldsymbol{W}_Q^{(i)}\boldsymbol{X}\right)\right],$$

where $\boldsymbol{W}_V^{(i)}, \boldsymbol{W}_K^{(i)}, \boldsymbol{W}_Q^{(i)} \in \mathbb{R}^{s\times d}$ and $\boldsymbol{W}_O^{(i)} \in \mathbb{R}^{d\times s}$ are the weight matrices, s is the head size, and $\sigma_S : \mathbb{R}^d \to \mathbb{R}^d$ is the softmax operator with output $[\sigma_S(\boldsymbol{x})]_i = e^{x_i}/\sum_{i=1}^d e^{x_i}, i \in [d]$ when acting on a vector $\boldsymbol{x}$. Here σ_S acts on the input matrix column-wise. For the sake of simplicity, similar to Yun et al. (2020a); Kim et al. (2023); Kajitsuka and Sato (2024), we excludes the layer normalization from the original definition of Vaswani et al. (2017) here.

A feedforward block $\boldsymbol{\mathcal{F}}_{FF}:\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n}$ with depth L and element-wise activation σ is defined by

$$\begin{aligned}\boldsymbol{\mathcal{F}}_0 &= \boldsymbol{X};\\ \boldsymbol{\mathcal{F}}_l &= \sigma(\boldsymbol{W}_l\boldsymbol{\mathcal{F}}_{l-1}+\boldsymbol{B}_l), \quad l\in\{1,2,\dots,L-1\};\\ \boldsymbol{\mathcal{F}}_{FF} &= \boldsymbol{W}_L\boldsymbol{\mathcal{F}}_{L-1}.\end{aligned}$$

where $\boldsymbol{W}_l\in\mathbb{R}^{n_l\times n_{l-1}}$ are the weight matrices, and $\boldsymbol{B}_l\in\mathbb{R}^{n_l\times n}$ are the bias matrices, n_l are the hidden dimensions. It can be observed that the feedforward block we define does not include skip connections. In comparison to the standard definition, we also allow the bias matrices $\boldsymbol{B}_l$ to have different columns. In the case where the columns of $\boldsymbol{B}_l$ are identical vectors $\boldsymbol{b}_l$: $\boldsymbol{B}_l=\boldsymbol{b}_l\mathbf{1}_{1\times n}$, we say that $\boldsymbol{\mathcal{F}}_{FF}$ is generated from a feedforward block $\boldsymbol{f}_{FF}$ acting on vectors, where $\boldsymbol{f}_{FF}:\mathbb{R}^d\to\mathbb{R}^d$ is defined in the following way:

$$\begin{aligned}\boldsymbol{f}_0 &= \boldsymbol{x};\\ \boldsymbol{f}_l &= \sigma(\boldsymbol{W}_l\boldsymbol{f}_{l-1}+\boldsymbol{b}_l), \quad l\in\{1,2,\dots,L-1\};\\ \boldsymbol{f}_{FF} &= \boldsymbol{W}_L\boldsymbol{f}_{L-1}.\end{aligned}$$

With these definitions, the Transformer $\boldsymbol{T}:\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n}$ is composed of alternating feedforward blocks and self-attention layers, with feedforward blocks at both the beginning and the end:

$$\boldsymbol{T}:=\boldsymbol{\mathcal{F}}_{FF}^{(N)}\circ\boldsymbol{\mathcal{F}}_{SA}^{(N)}\circ\boldsymbol{\mathcal{F}}_{FF}^{(N-1)}\circ\cdots\circ\boldsymbol{\mathcal{F}}_{FF}^{(1)}\circ\boldsymbol{\mathcal{F}}_{SA}^{(1)}\circ\boldsymbol{\mathcal{F}}_{FF}^{(0)}.$$

2.2 Kolmogorov–Arnold Representation Theorem

Kolmogorov-Arnold Superposition Theorem (KST), first proposed by Kolmogorov in 1957 (Kolmogorov, 1957), states that any continuous multivariate function can be represented as a superposition of continuous functions of fewer variables. This theorem provides a fundamental insight into the representation of functions in high-dimensional spaces, offering a way to simplify complex functions by breaking them down into simpler components. It is worth mentioning that KST is closely related to Hilbert's Thirteenth Problem.

Below we present the original KST:

Proposition 1 (KST) *There exist univariate continuous functions $\psi_{p,q}$ such that for any continuous function $f:[0,1]^d\to\mathbb{R}$, there exist univariate continuous functions g_q such that*

$$f(x_1,\dots,x_d)=\sum_{q=0}^{2d}g_q\left(\sum_{p=1}^{d}\psi_{p,q}(x_p)\right).$$

Since the proposal of KST, a great number of variants have emerged. In 2021, Hieber established a type of KST using the technique of space-filling curves, which can transfer the smoothness properties of the target function to the outer function (Schmidt-Hieber, 2021). This discovery enables researchers to apply KST to investigate the approximation properties of FNNs. Our work is also built upon his result.

Definition 2 (Hölder continuous function) *Let $\beta\le 1, Q\in\mathbb{R}$. $f:[0,1]^d\to\mathbb{R}$ is called (β,Q)-Hölder continuous function if for any $\boldsymbol{x},\boldsymbol{y}\in[0,1]^d$, there holds*

$$|f(\boldsymbol{x})-f(\boldsymbol{y})|\le Q\|\boldsymbol{x}-\boldsymbol{y}\|_\infty^\beta. \tag{1}$$

where $\|\cdot\|_\infty$ is the infty norm of vectors. $\mathcal{H}_Q^\beta\left([0,1]^d,\mathbb{R}\right)$ denotes the class that consisting all the (β,Q)-Hölder continuous function $f:[0,1]^d\to\mathbb{R}$.

Definition 3 *Let $\mathcal{C}$ be the Cantor set. $\mathcal{H}_Q^{\beta}\left(\mathcal{C},\mathbb{R}\right)$ is defined in a manner similar to Definition 2.*

Proposition 4 (Schmidt-Hieber (2021), Theorem 2) *Let $d \in \mathbb{N}_{\geq 1}$. There exists a monotone function $\phi : [0,1] \to \mathcal{C}$ such that for any function $f \in \mathcal{H}_Q^{\beta}\left([0,1]^d,\mathbb{R}\right)$, we can find a function $g \in \mathcal{H}_{2^{\beta}Q}^{\frac{\beta \log 2}{d \log 3}}\left(\mathcal{C},\mathbb{R}\right)$ such that*

$$f\left(x_1,\ldots,x_d\right) = g\left(3\sum_{p=1}^{d} 3^{-p}\phi\left(x_p\right)\right).$$

In Schmidt-Hieber (2021), the explicit form of $\phi(x)$ is provided. For $x = [0.a_1^x a_2^x \ldots]_2 \in [0,1]$,

$$\phi(x) = \sum_{j=1}^{\infty} 2a_j^x 3^{-1-d(j-1)}.$$

An approximation of $\phi(x)$ with the precision parameter K is given by

$$\phi_K(x) := \sum_{j=1}^{K} 2a_j^x 3^{-1-d(j-1)},$$

which will be used in our construction.

Remark 5 *Proposition 4 achieves the transfer of smoothness from the target function f to the outer function g at the cost of discontinuities in the inner function ϕ. All discontinuity points of ϕ consist of the dyadic rationals within the interval $[0,1]$. Dyadic rationals refer to those numbers possessing finite binary expansions. Their binary expansions are not unique; there is one finite and one infinite representation of each dyadic rational other than 0. For instance, for the dyadic rational $1/2$, the finite binary representation is $[0.1]_2$, while the infinite binary representation is $[0.0111\cdots]_2$. The dyadic rationals are the only numbers whose binary expansions are not unique. For further discussion on dyadic rationals, readers may refer to Ko (2012).*

At these discontinuity points, ϕ can take on two possible values (since dyadic rationals admit exactly two binary representations), and either choice is mathematically valid. Once a specific value assignment for ϕ at its discontinuity points is prescribed, ϕ becomes well-defined. Under the construction in Schmidt-Hieber (2021), the resulting function Φ^{-1} is Hölder continuous (where $\Phi := 3\sum_{p=1}^{d} 3^{-p}\phi(x_p)$), and consequently, the composition $g = f \circ \Phi^{-1}$ is also Hölder continuous.

In this work, we construct FNNs to implement ϕ_K, the truncation of ϕ. Specifically, Lemma 15 employs continuous ReLU-FNNs for implementation, where the discontinuity points of ϕ_K are contained within the flawed region $\Omega^{(flaw)}$. In Lemma 16, discontinuous floor-ReLU-FNNs are constructed for implementation. According to the proof process, it can be observed that in this case, the definition of ϕ_K at discontinuity points (dyadic rationals) adopts the finite binary representation of these points.

2.3 Notations

- $\mathbb{N}_{\geq 1}$ is the set consisting of all positive integers. $[N] = \{1,2,\ldots,N\}$.
- The definition of $\mathcal{H}_Q^{\beta}\left([0,1]^{d\times n},\mathbb{R}\right)$ is the same as Definition 2, except that the vector infinity norm in (1) is replaced by the element-wise matrix infinity norm. For $\boldsymbol{f} : [0,1]^{d\times n} \to \mathbb{R}^{d\times n}$, we say that $\boldsymbol{f} \in \mathcal{H}_Q^{\beta}([0,1]^{d\times n},\mathbb{R}^{d\times n})$ if $f_{rs} \in \mathcal{H}_Q^{\beta}([0,1]^{d\times n},\mathbb{R})$ for all $r \in [n], s \in [d]$.

- Let $p \in [1, \infty)$. The distances d_∞ and d_p between two functions $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ are defined in the follwing manners:

$$d_\infty(\boldsymbol{f}, \boldsymbol{g}) := \max_{\boldsymbol{X}\in[0,1]^{d\times n}} \max_{r\in[d], s\in[n]} |g_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})|,$$

$$d_p(\boldsymbol{f}, \boldsymbol{g}) := \left(\int_{\boldsymbol{X}\in[0,1]^{d\times n}} \sum_{r\in[d], s\in[n]} |g_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})|^p d\boldsymbol{X} \right)^{1/p}.$$

- Let $\sigma_R(x) = \begin{cases} x, & x \geq 0; \\ 0, & x < 0 \end{cases}$, $\sigma_F(x) = \lfloor x \rfloor$. Let $\sigma_{RC}(x) = \frac{1}{\alpha + x}$ with α being any transcendental number (A transcendental number is a complex number that is not a root of any polynomial equation with rational coefficients). Let σ_{NP} be any real analytic function that is non-polynomial in some interval. Let σ_P be any periodic function.
- Let $\mathcal{FF}(L, W, A)$ be the set that contains all feedforward neural network functions with depth L, width W and activation functions in set A.

3. Transformer Can Overcome the Curse of Dimensionality

3.1 Main Results

The following two theorems are the main results of this paper. Here $f_{\max}, g_{\max}$ and $g_{\min}$ are defined by

$$f_{\max} := \max_{1\leq r\leq d, 1\leq s\leq n} \max_{\boldsymbol{x}\in[0,1]^d} f_{rs}(\boldsymbol{x}),$$

$$g_{\max} := \max_{1\leq r\leq d, 1\leq s\leq n} \max_{x\in\mathcal{C}} g_{rs}(x),$$

$$g_{\min} := \min_{1\leq r\leq d, 1\leq s\leq n} \min_{x\in\mathcal{C}} g_{rs}(x),$$

where $\{f_{rs}\}$ are the components of $\boldsymbol{f} \in \mathcal{H}_Q^\beta\left([0,1]^{d\times n}, \mathbb{R}^{d\times n}\right)$, $\{g_{rs}\}$ are the outer functions in Proposition 4 for $\{f_{rs}\}$.

Theorem 6 *Let $\boldsymbol{f} \in \mathcal{H}_Q^\beta\left([0,1]^{d\times n}, \mathbb{R}^{d\times n}\right)$. For any $\epsilon > 0$, there exist feedforward blocks $\boldsymbol{\mathcal{F}}_{FF}^{(sr)} : \mathbb{R}^{d\times n} \to \mathbb{R}^{2d\times n}, \widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{d\times n}$ and a self-attention layer $\boldsymbol{\mathcal{F}}_{SA} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{2d\times n}$ such that for the transformer $\boldsymbol{T} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ defined by*

$$\boldsymbol{T} := \widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} \circ \boldsymbol{\mathcal{F}}_{SA} \circ \boldsymbol{\mathcal{F}}_{FF}^{(sr)},$$

there holds

$$d_\infty(\boldsymbol{T}, \boldsymbol{f}) \leq \epsilon.$$

$\boldsymbol{\mathcal{F}}_{FF}^{(sr)}$ is a feedforward block with depth $\left\lceil \frac{1}{\beta} \log_2 \frac{2^{1-\beta}Q}{\epsilon} \right\rceil + 3$, width $3dn$ and activation functions being ReLU and floor. The form of $\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)}$ can be one of the following:

(1) Its activation functions are ReLU and floor, its depth is $7dn - 2$ and its width is $d\left(2\left\lceil n\left(\frac{2^{1-\beta}Q}{\epsilon}\right)^{2/\beta}\right\rceil + 2\right)\left\lceil \log_2 \frac{2(g_{\max} - g_{\min})}{\epsilon} \right\rceil$;

(2) *Its activation functions are ReLU, floor and* 2^x*, its depth is* 6 *and its width is* $\left\lceil 3d \log_2 \frac{2(g_{\max}-g_{\min})}{\epsilon} \right\rceil$;

(3) *Its activation functions are ReLU, floor and* σ_{NP}*, its depth is* 3 *and its width is* $3d$;

(4) *Its activation functions are ReLU, floor and* σ_{RC}*, its depth is* 3 *and its width is* $3d$.

Theorem 7 *Let* $\boldsymbol{f} \in \mathcal{H}_Q^\beta\left([0,1]^{d\times n}, \mathbb{R}^{d\times n}\right)$*. Let* $p \in [1,\infty)$*. For any* $\epsilon > 0$*, there exist feedforward blocks* $\boldsymbol{\mathcal{F}}_{FF}^{(sr)} : \mathbb{R}^{d\times n} \to \mathbb{R}^{2d\times n}, \widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{d\times n}$ *and a self-attention layer* $\boldsymbol{\mathcal{F}}_{SA} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{2d\times n}$ *such that for the transformer* $\boldsymbol{T} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ *defined by*

$$\boldsymbol{T} := \widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} \circ \boldsymbol{\mathcal{F}}_{SA} \circ \boldsymbol{\mathcal{F}}_{FF}^{(sr)},$$

there holds

$$d_p(\boldsymbol{T}, \boldsymbol{f}) \leq \epsilon.$$

$\boldsymbol{\mathcal{F}}_{FF}^{(sr)}$ *is a feedforward block with depth*

$$2\max\left\{\left\lceil \frac{1}{\beta}\log_2 \frac{2^{1/p}d^{2/p}n^{2/p}(B_\sigma + f_{\max})}{\epsilon}\right\rceil, \left\lceil \frac{1}{\beta}\log_2 \frac{2^{1-\beta}(2dn)^{1/p}Q}{\epsilon}\right\rceil\right\},$$

width $4dn$ *and activation functions being ReLU. The form of* $\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)}$ *can be one of the following:*

(1) *Its activation functions are ReLU, sine and* 2^x*, its depth is* 4 *and its width is* $2d\left\lceil \log_2 \frac{2(2dn)^{1/p}(g_{\max}-g_{\min})}{\epsilon}\right\rceil$.

(2) *Its activation functions are ReLU, cosine and* 3^x*, its depth is* 5 *and its width is* $d\left\lceil \log_2 \frac{2(2dn)^{1/p}(g_{\max}-g_{\min})}{\epsilon}\right\rceil$.

(3) *Its activation functions are* σ_P *and* σ_{NP}*, its depth is* 3 *and its width is* $3d$.

(4) *Its activation functions are* σ_P *and* σ_{RC}*, its depth is* 3 *and its width is* $3d$.

Remark 8 *The size of* $\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)}$ *presented in Theorem 6 gradually decreases. We can observe that the parameter number of (1) is* $\mathcal{O}(\log\frac{1}{\epsilon})$*; whereas the parameter number of (4) is* $\mathcal{O}(1)$*. The same applies to Theorem 7.*

Remark 9 *Based on the proofs in Section 3.3, we know that Theorems 6 and 7 only present a subset of Transformers that can overcome the curse of dimensionality. Our goal here is to limit the total number of activation function types in the feedforward layers to no more than three. In fact, if more activation functions are allowed, according to Lemmas 12, 13 and 14, we can generate more Transformers that does not suffer from the curse of dimensionality.*

3.2 Basic Idea of Our Construction

In this section, we illustrate the basic idea of how we construct the transformer by considering the case when $d = 2, n = 3$. In this case,

$$\boldsymbol{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix} \in \mathbb{R}^{2\times 3}$$

and the target function $\boldsymbol{f} \in \mathcal{H}_Q^\beta\left([0,1]^{2\times 3}, \mathbb{R}^{2\times 3}\right)$ can be written in the following way:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_{11}(\boldsymbol{x}) & f_{12}(\boldsymbol{x}) & f_{13}(\boldsymbol{x}) \\ f_{21}(\boldsymbol{x}) & f_{22}(\boldsymbol{x}) & f_{23}(\boldsymbol{x}) \end{pmatrix},$$

where $f_{rs} \in \mathcal{H}_Q^\beta\left([0,1]^{2\times 3}, \mathbb{R}\right)$ for $r = 1,2; s = 1,2,3$.

Proposition 4, the Kolmogorov–Arnold representation theorem, enables us to find $g_{rs} \in \mathcal{H}_{2^\beta Q}^{\frac{\beta \log 2}{6 \log 3}}(\mathcal{C}, \mathbb{R})$ such that for $r = 1,2; s = 1,2,3$,

$$f_{rs}(\boldsymbol{x}) = g_{rs}\left(3\sum_{p=1}^{2}\sum_{q=1}^{3} a_{pq}\phi(x_{pq})\right).$$

where $a_{pq} := \frac{1}{3^{3(p-1)+q}}$.

Let $\phi_K(x)$ be an approximant of $\phi(x)$:

$$\phi_K(x) := 2\sum_{j=1}^{K} a_j^x 3^{-6(j-1)-1}$$

and

$$a(\boldsymbol{x}) := \sum_{p=1}^{2}\sum_{q=1}^{3} a_{pq}\phi_K(x_{pq}),$$
$$b(\boldsymbol{x}) := 3^{6K+1}a(\boldsymbol{x}) + 1.$$

It can be seen that $g_{rs}\left(3a(\boldsymbol{x})\right)$ is an approximant of $f_{rs}(\boldsymbol{x})$.

Below is a rough process of our construction in this paper. Here, steps one to two are implemented using a feedforward block; step three is implemented using a self-attention layer; steps four to six are implemented using another feedforward block.

$$\begin{aligned}
&\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix} \\
\Longrightarrow &\begin{pmatrix} x_{11} & x_{12}+2 & x_{13}+4 \\ x_{21} & x_{22}+2 & x_{23}+4 \end{pmatrix} \\
\Longrightarrow &\begin{pmatrix} \sum_p a_{p1}\phi_K(x_{p1}) & \sum_p a_{p2}\phi_K(x_{p2}) & \sum_p a_{p3}\phi_K(x_{p3}) \\ \sum_p a_{p1}\phi_K(x_{p1}) & \sum_p a_{p2}\phi_K(x_{p2}) & \sum_p a_{p3}\phi_K(x_{p3}) \end{pmatrix} \\
\Longrightarrow &\begin{pmatrix} a(\boldsymbol{x}) & a(\boldsymbol{x}) & a(\boldsymbol{x}) \\ a(\boldsymbol{x}) & a(\boldsymbol{x}) & a(\boldsymbol{x}) \end{pmatrix} \\
\Longrightarrow &\begin{pmatrix} b(\boldsymbol{x}) & b(\boldsymbol{x}) & b(\boldsymbol{x}) \\ b(\boldsymbol{x}) & b(\boldsymbol{x}) & b(\boldsymbol{x}) \end{pmatrix}
\end{aligned}$$

$$\Longrightarrow \begin{pmatrix} b(\boldsymbol{x}) & b(\boldsymbol{x})+3^{6K} & b(\boldsymbol{x})+2\cdot 3^{6K} \\ b(\boldsymbol{x}) & b(\boldsymbol{x})+3^{6K} & b(\boldsymbol{x})+2\cdot 3^{6K} \end{pmatrix}$$
$$\Longrightarrow \begin{pmatrix} g_{11}(3a(\boldsymbol{x})) & g_{12}(3a(\boldsymbol{x})) & g_{13}(3a(\boldsymbol{x})) \\ g_{21}(3a(\boldsymbol{x})) & g_{22}(3a(\boldsymbol{x})) & g_{23}(3a(\boldsymbol{x})) \end{pmatrix}$$

Now let's explain each of these steps. First, we are going to explain steps one and two. It is difficult to directly transform $\boldsymbol{x}$ into the matrix in the third row. This is because to achieve this, it is necessary to use three different column functions to map each column (a two-dimensional vector) to a new two-dimensional vector. Note that these column functions are each unique pairwise, as can be understood from the fact that the combination coefficients for the first column are a_{p1} while for the second column they are a_{p2}. Since ϕ_K can be (approximately) implemented using a feedforward neural network function, each column function can also be (approximately) implemented using a feedforward neural network function. However, because the three column functions are different, it is not possible to use the same feedforward neural network function to simultaneously implement the transformation of all three columns. To address this issue, we have developed a technique here: by applying different translations to the three columns, at the cost of broadening the domain of the column functions, we transform the problem of implementing three distinct column functions into that of implementing a single column function. Therefore, the role of step one is to perform a translation on each column, while step two involves implementing a specific column function.

Step three involves summing the matrix columns, an operation that can be implemented using a self-attention layer.

Step four is a linear mapping step designed to ensure that the points to be memorized in the memory task in step six are all integers. This allows us to utilize the results of integer memorization from the feedforward neural network (Lemma 17,19,20). Step five is another translation step, with a purpose similar to the first step: through this translation step, we transform the problem of implementing three different column functions into implementing a single column function. Step six involves two memorization problems concerning up to $3\cdot 3^{6K}$ integers: given $r\in\{1,2\}$, find a feedforward neural network function $f_{FF,r}^{(mmr)}$ such that for any $\boldsymbol{x}\in[0,1]^d$ and $s\in\{1,2,3\}$, there holds

$$f_{FF,r}^{(mmr)}\left(b(\boldsymbol{x})+(s-1)\cdot 3^{6K}\right)=g_{rs}\left(3a(\boldsymbol{x})\right).$$

This is because the set

$$\Lambda := \{b(\boldsymbol{x})+(s-1)\cdot 3^{6K} : \boldsymbol{x}\in[0,1]^{2\times 3}, s\in\{1,2,3\}\}$$

is finite and $\Lambda\subset\left\{1,2,\dots,3\cdot 3^{6K}\right\}$.

Remark 10 *The cardinality of* Λ *is actually* $|\Lambda|=3\cdot 2^{6K}$*, which we here magnify to* $3\cdot 3^{6K}$*. This manipulation has a negligible effect on the orders of the size of the transformers in Theorems 6 and 7.*

Remark 11 *From the process described above, we can see that the summation required in the KST is divided into two parts in its construction. The summation over rows is achieved by the feedforward layers, while the summation over columns is accomplished by the self-attention mechanism.*

As a model that accepts multiple input tokens, to achieve the approximation of Hölder continuous functions, the Transformer model must implement some form of interaction between tokens, that is, between columns. Many previous studies on the approximation ability of Transformers have involved complex inter-column interactions, whereas our construction offers a simple and intuitive *perspective: even using only summation across columns, without any more intricate operations, suffices to approximate Hölder continuous functions. This also indicates that KST provides an excellent framework for studying the expressive power of Transformers. The column-wise summation in KST cannot be implemented by feedforward blocks. This is because feedforward blocks can only operate on the rows of the input matrix and cannot process columns (since the weight matrices in feedforward blocks perform left-multiplication on the input independent variables). Therefore, the self-attention structure is indispensable and cannot be replaced by feedforward blocks.*

3.3 Proof of Theorems 6 and 7

We define the inner matrix as the matrix in step five in section 3.2. More generally, given

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & \cdots & x_{dn} \end{pmatrix} \in [0,1]^{d\times n},$$

define the inner matrix $\boldsymbol{Z} := \boldsymbol{Z}(\boldsymbol{X}) \in \mathbb{R}^{d\times n}$ by

$$\boldsymbol{Z}_{:,s} = \left[\sum_{q=1}^{n}\sum_{p=1}^{d} \frac{3^{Kdn+1}}{3^{(p-1)s+q}}\phi_K(x_{pq}) + 1 + (s-1)\cdot 3^{Kdn}\right]\mathbf{1}_{d\times 1}$$

for $s \in [n]$. With this definition, the proof of Theorems 6 and 7 can be divided into two parts: one part proves that for (almost) all $\boldsymbol{X} \in [0,1]^{d\times n}$, there exist a feedforward block $\boldsymbol{\mathcal{F}}_{FF}^{(sr)}$, a self-attention layer $\boldsymbol{\mathcal{F}}_{SA}$, and a linear transformation layer $\boldsymbol{\mathcal{A}}$ such that

$$\boldsymbol{Z} = \boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{F}}_{SA} \circ \boldsymbol{\mathcal{F}}_{FF}^{(sr)}(\boldsymbol{X});$$

the other part proves that for a given $\boldsymbol{Z}$, there exists a feedforward block $\boldsymbol{\mathcal{F}}_{FF}^{(mmr)}$ such that

$$\boldsymbol{\mathcal{F}}_{FF}^{(mmr)}(\boldsymbol{Z}) \approx \boldsymbol{f}(\boldsymbol{X}).$$

These two parts correspond respectively to steps one to five and step six in section 3.2. The first part is guaranteed by Lemma 12 below.

Lemma 12 *Let $K \in \mathbb{N}_{\geq 1}$. Let $p \in [1,\infty)$. There exist a feedforward block $\boldsymbol{\mathcal{F}}_{FF}^{(sr)} : \mathbb{R}^{d\times n} \to \mathbb{R}^{2d\times n}$, a self-attention layer $\boldsymbol{\mathcal{F}}_{SA} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{2d\times n}$ and a linear transformation layer $\boldsymbol{\mathcal{A}} : \mathbb{R}^{2d\times n} \to \mathbb{R}^{d\times n}$ such that*

$$\boldsymbol{Z} = \boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{F}}_{SA} \circ \boldsymbol{\mathcal{F}}_{FF}^{(sr)}(\boldsymbol{X}). \tag{2}$$

The form of $\boldsymbol{\mathcal{F}}_{FF}^{(sr)}$ can be one of the following:

(1) Its activation function is ReLU, its depth is $2K$ and its width is $4dn$.

(2) Its activation functions are ReLU and floor, its depth is $K+3$ and its width is $3dn$.

For the first case, (2) holds for $\boldsymbol{X} \in \{[0,1]\backslash\Omega^{(flow)}\}^{d\times n}$; for the second case, (2) holds for $\boldsymbol{X} \in [0,1]^{d\times n}$. Here $\Omega^{(flaw)} \subset [0,1]$ is some region with the Lebesgue measure not greater than $2^{-K\beta p}$.

The second part is guaranteed by Lemmas 13 and 14 below, where the former is based on binary memorization, and the latter is based on the memorization with labels of arbitrary real numbers.

Lemma 13 *Let $K, H \in \mathbb{N}_{\geq 1}$. There exists a feedforward neural network function $\boldsymbol{f}_{FF}^{(mmr)} : \mathbb{R}^d \to \mathbb{R}^d$ such that for $\widetilde{\boldsymbol{f}}(\boldsymbol{X}) := \boldsymbol{\mathcal{F}}_{FF}^{(mmr)}(\boldsymbol{Z}(\boldsymbol{X}))$, there holds*

$$|\widetilde{f}_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})| \leq \frac{g_{\max} - g_{\min}}{2^H} + \frac{2^\beta Q}{2^{(K+2)\beta}},$$

for any $\boldsymbol{X} \in [0,1]^d, r \in [d], s \in [n]$. $\boldsymbol{f}_{FF}^{(mmr)}$ can be in one of the following feedforward neural network classes:

(1) $\mathcal{FF}\left(7L-2, \left(2\left\lceil (n3^{Kdn})^{1/L}\right\rceil + 2\right) dH, \{ReLU, floor\}\right)$, where L can be any positive integer number;

(2) $\mathcal{FF}(6, 3dH, \{ReLU, floor, 2^x\})$;

(3) $\mathcal{FF}(4, 2dH, \{ReLU, sine, 2^x\})$;

(4) $\mathcal{FF}(5, dH, \{ReLU, cosine, 3^x\})$.

Lemma 14 *Let $K \in \mathbb{N}_{\geq 1}$. There exists a feedforward neural network function $\boldsymbol{f}_{FF}^{(mmr)} : \mathbb{R}^d \to \mathbb{R}^d$ such that for $\widetilde{\boldsymbol{f}}(\boldsymbol{X}) := \boldsymbol{\mathcal{F}}_{FF}^{(mmr)}(\boldsymbol{Z}(\boldsymbol{X}))$, there holds*

$$|\widetilde{f}_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})| \leq \delta + \frac{2^\beta Q}{2^{(K+2)\beta}},$$

for any $\boldsymbol{X} \in [0,1]^d, r \in [d], s \in [n]$. Here $\delta > 0$ can be arbitrarily small. $\boldsymbol{f}_{FF}^{(mmr)}$ can be in one of the following feedforward neural network classes:

(1) $\mathcal{FF}(3, d, \{\sigma_P, \sigma_{NP}\})$;

(2) $\mathcal{FF}(3, d, \{\sigma_P, \sigma_{RC}\})$.

3.3.1 Proof of Theorem 6

(1) and (2) arise from (2) of Lemma 12 and (1)(2) of Lemma 13. (3) and (4) arise from (2) of Lemma 12 and (1)(2) of Lemma 14. Note that here the general periodic function $\sigma_P(x)$ in Lemma 14 is replaced by a specific periodic function with period 1 and equaling to x on $[0,1)$. This function can be implemented by a ReLU-floor feedforward network function with depth 2 and width 3:

$$\sigma_R(x) + \sigma_R(-x) - \sigma_F(x).$$

$\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)}$ is defined by

$$\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} := \boldsymbol{\mathcal{F}}_{FF}^{(mmr)} \circ \boldsymbol{\mathcal{A}},$$

where $\boldsymbol{\mathcal{A}}$ is from Lemma 12 and $\boldsymbol{\mathcal{F}}_{FF}^{(mmr)}$ is from Lemma 13 or Lemma 14. Parameters in these lemmas are set to be

$$\begin{aligned} H &= \left\lceil \log_2 \frac{2(g_{\max} - g_{\min})}{\epsilon} \right\rceil, \\ K &= \left\lceil \frac{1}{\beta} \log_2 \frac{2^{1-\beta} Q}{\epsilon} \right\rceil, \\ L &= dn. \end{aligned}$$

3.3.2 Proof of Theorem 7

(1) and (2) arise from (1) of Lemma 12 and (3)(4) of Lemma 13. (3) and (4) arise from (1) of Lemma 12 and (1)(2) of Lemma 14. $\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)}$ is defined by

$$\widetilde{\boldsymbol{\mathcal{F}}}_{FF}^{(mmr)} := \boldsymbol{\mathcal{F}}_{FF}^{(mmr)} \circ \boldsymbol{\mathcal{A}},$$

where $\boldsymbol{\mathcal{A}}$ is from Lemma 12 and $\boldsymbol{\mathcal{F}}_{FF}^{(mmr)}$ is from Lemma 13 or Lemma 14.

To bound the L^p distance of $\boldsymbol{T}$ and $\boldsymbol{f}$, we decompose the whole integral into two parts:

$$\begin{aligned}
&[d_p(\boldsymbol{T}, \boldsymbol{f})]^p \\
&= \int_{\boldsymbol{X}\in[0,1]^d} \sum_{r\in[d],s\in[n]} |T_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})|^p d\boldsymbol{X} \\
&= \int_{\boldsymbol{X}\in\Omega_1} \sum_{r\in[d],s\in[n]} |T_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})|^p d\boldsymbol{X} + \int_{\boldsymbol{X}\in\Omega_2} \sum_{r\in[d],s\in[n]} |T_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})|^p d\boldsymbol{X},
\end{aligned}$$

where $\Omega_1 := \{[0,1] \setminus \Omega^{(flow)}\}^{d\times n}$ denotes the region where T_{rs} can effectively approximate f_{rs}, whereas $\Omega_2 := [0,1]^{d\times n} \setminus \{[0,1] \setminus \Omega^{(flow)}\}^{d\times n}$ represents the region where T_{rs} fails to approximate f_{rs} well.

Lemma 13 enables us to derive an upper bound for the first term(the cases when adopting Lemma 14 can be handled almost the same and hence we omit this part):

$$dn\left(\frac{g_{\max} - g_{\min}}{2^H} + \frac{2^\beta Q}{2^{(K+2)\beta}}\right)^p.$$

By our construction (see Lemma 13 and Lemma 14 for details), the range of T_{rs} is bounded by some constants B_σ, therefore the integrand of the second term is simply bounded by $dn(B_\sigma + f_{\max})^p$. Furthermore, since $m\left(\Omega^{(flow)}\right) \le 2^{-K\beta p}$, by Bernoulli's inequality we can derive a bound for the measure of the region of the second term:

$$\begin{aligned}
m\left([0,1]^d\backslash\{[0,1]\backslash\Omega^{(flow)}\}^{d\times n}\right) &= 1 - \left(1 - m\left(\Omega^{(flow)}\right)\right)^{dn} \\
&\le dn \cdot m\left(\Omega^{(flow)}\right) \le dn2^{-K\beta p}.
\end{aligned}$$

Therefore the second term is bounded by

$$2^{-K\beta p} d^2 n^2 (B_\sigma + f_{\max})^p.$$

By far we obtain

$$[d_p(\boldsymbol{T}, \boldsymbol{f})]^p \le dn\left(\frac{g_{\max} - g_{\min}}{2^H} + \frac{2^\beta Q}{2^{(K+2)\beta}}\right)^p + \frac{d^2 n^2 (B_\sigma + f_{\max})^p}{2^{K\beta p}}.$$

By setting the parameters

$$\begin{aligned}
K &= \max\left\{\left\lceil \frac{1}{\beta}\log_2 \frac{2^{1/p} d^{2/p} n^{2/p} (B_\sigma + f_{\max})}{\epsilon}\right\rceil, \left\lceil \frac{1}{\beta}\log_2 \frac{2^{1-\beta}(2dn)^{1/p} Q}{\epsilon}\right\rceil\right\}, \\
H &= \left\lceil \log_2 \frac{2(2dn)^{1/p}(g_{\max} - g_{\min})}{\epsilon}\right\rceil,
\end{aligned}$$

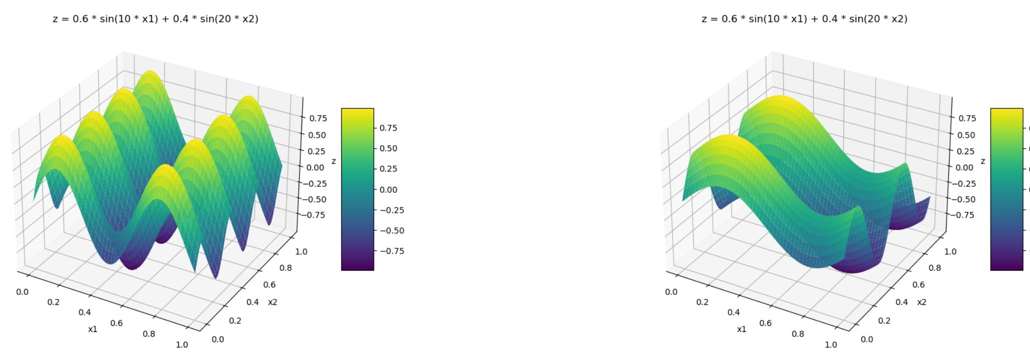
we arrive the desired accuracy ϵ.

4. Experiments

In this section, we conduct simple experiments testing the stability of the proposed Transformers during training. We consider using our Transformers to approximate the objective function

$$\boldsymbol{f}(x_1, x_2) := (0.6\sin(10x_1) + 0.4\sin(20x_2), 0.4\sin(6x_1) + 0.6\sin(12x_2)).$$

The 3D plot of the two components of this function is shown in Figure 1.



(a) $f_1(x_1, x_2) = 0.6\sin(10x_1) + 0.4\sin(20x_2)$ (b) $f_2(x_1, x_2) = 0.4\sin(6x_1) + 0.6\sin(12x_2)$

Figure 1: Target function

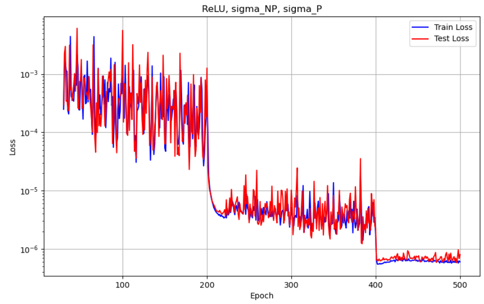
The training dataset consists of 10,000 samples, and the test dataset consists of 2,000 samples. Both are uniformly distributed on $[0, 1]^2$. The Transformer consists of 2 feedforward blocks $\boldsymbol{\mathcal{F}}_1$ and $\boldsymbol{\mathcal{F}}_2$ and one self-attention layer $\boldsymbol{\mathcal{F}}_{SA}$. The width of $\boldsymbol{\mathcal{F}}_1$ and $\boldsymbol{\mathcal{F}}_2$ is 50, with a depth of 5 for $\boldsymbol{\mathcal{F}}_1$ and a depth of 3 for $\boldsymbol{\mathcal{F}}_2$. The number of heads in $\boldsymbol{\mathcal{F}}_{SA}$ is 5. Dropout is applied to $\boldsymbol{\mathcal{F}}_1$. The mean squared error (MSE) is used as the loss function. Training is performed using the Adam algorithm Kingma and Ba (2015) with a batch size of 64. The training runs for 500 epochs. The initial learning rate is set to 0.001, which decays by a factor of 0.1 every 200 epochs. Gradient clipping is applied during the training process. Figure 2 shows the descent curves of training loss and test loss during training under several combinations of activation functions in $\boldsymbol{\mathcal{F}}_1$ and $\boldsymbol{\mathcal{F}}_2$. These experiments demonstrate that the Transformer we proposed has a certain level of stability during training.

5. Conclusions

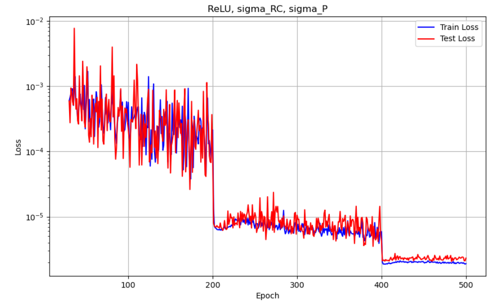
This paper investigates the approximation of the Hölder continuous function class by Transformers and constructs several Transformers that can overcome the curse of dimensionality. These Transformers consist of one self-attention layer with one head and the softmax function as the activation function, along with several small-sized feedforward layers. These results demonstrate that Transformers have a strong expressive capability. The construction in this paper is based on the Kolmogorov-Arnold Superposition Theorem and does not require the concept of contextual mapping. Additionally, the translation technique proposed in this paper helps to apply the previous approximation results of feedforward neural networks to Transformer research.

There are still many research topics related to Transformers worth exploring in the future. Below, we outline two possible directions.

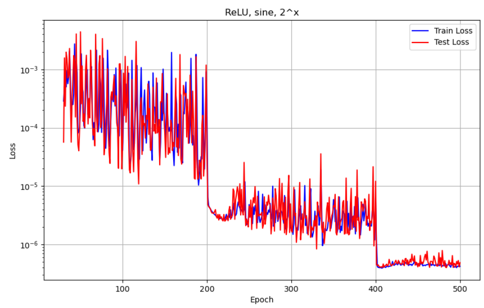
Assuming the target function is Hölder continuous provides a natural starting point for studying the approximation capabilities of Transformers. When considering the approximation of Transformers



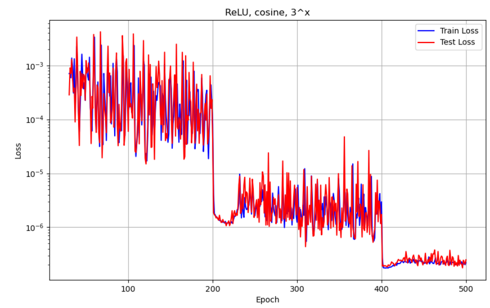
(a) ReLU, σ_{NP}, σ_P



(b) ReLU, σ_{RC}, σ_P



(c) ReLU, sine, 2^x



(d) ReLU, cosine, 3^x

Figure 2: Training and test losses during training

in general continuous function spaces $C^s(\Omega)$, Sobolev spaces $W^{s,p}(\Omega)$, or Besov spaces $B^s_{p,q}(\Omega)$, the framework presented in this paper is no longer applicable. This is because, to the best of our knowledge, there are currently no generalizations of KST for these specific spaces. A common approach to studying the neural network approximation capabilities in these function spaces is to first approximate functions in these spaces using piecewise polynomials, and then use neural networks to approximate the piecewise polynomials. This methodology has been frequently employed in studies on the approximation power of FNNs (Lu et al., 2021; Gühring and Raslan, 2021; Siegel, 2023; Yang, 2025). However, how to use Transformers to approximate polynomial functions remains a challenging problem. In fact, this is precisely part of our ongoing work.

Neural networks for solving least squares regression problems represent an important application in the field. Several works have demonstrated that FNNs can achieve the minimax optimal rate in this task (Schmidt-Hieber, 2020; Jiao et al., 2023b; Yang and Zhou, 2024b; Nakada and Imaizumi, 2020). It is worth considering whether Transformers can also attain the minimax optimal rate in this task. To this end, a trade-off between the approximation error and generalization error of Transformers needs to be carefully balanced (Yang and Zhou, 2024a,b; Nakada and Imaizumi, 2020). The current work focuses primarily on approximation error. Unconventional activation functions such as σ_P, σ_{NP}, σ_{RC}, and the floor function exhibit strong expressive power in our construction; however, this also implies high complexity, leading to large generalization error. For this reason, we believe that the convergence rate of the Transformer proposed in this work for solving least squares regression may not achieve the minimax optimal rate. If the activation functions in the feedforward blocks of the Transformer are restricted to ReLU (i.e., the standard setting), it would be worthwhile to explore whether the minimax optimal rate can be derived.

Acknowledgments

The work of Yuling Jiao was supported in part by the National Key Research and Development Program of China under Grant 2024YFA1014202, in part by the Fundamental Research Funds for the Central Universities (Grant No. 2042025kf0023) and in part by the National Natural Science Foundation of China under Grant 12371441, 12526216. The work of Yang Wang was supported in part by Hong Kong Innovation and Technology Fund (PRP/068/21FX). We thank the editor and reviewers for their valuable feedback on our manuscript.

Overview of Appendices

Appendix A provides a detailed comparison of our work with related works. Appendix B constructs the FNNs to implement the inner function ϕ_K of KST, which forms the basis for proving Lemma 12. Appendix C presents some memorization results of feedforward neural networks, which serve as the foundation for proving Lemmas 13 and 14. Appendices D, E, and F provide the proofs for Lemmas 12, 13, and 14, respectively.

Appendix A. Comparison of Related Work

In Table 2, we have listed our results along with a comparison to some related works.

Appendix B. Implementation of ϕ_K

The following lemma suggests the existence of a ReLU-feedforward neural network function $f_{FF}^{(R-pi)}$ that can realize a piecewise function generated by ϕ_K at almost all $x \in [0, 2n-1]$.

Lemma 15 *Let $p \in [1, \infty)$. There exists a ReLU-feedforward neural network function $f_{FF}^{(R-pi)} : \mathbb{R} \to \mathbb{R}$ with depth $2K$ and width $4n$ such that for $x \in [2q, 2q+1] \backslash (\Omega^{(flow)} + 2q), q \in \{0, 1, \ldots, n-1\}$, there holds*

$$f_{FF}^{(R-pi)}(x) = \frac{1}{3^{q-1}} \phi_K(x - 2q) + \frac{9}{2} - \frac{1}{2} \frac{1}{3^{q-2}}.$$

Here $\Omega^{(flaw)} \subset [0, 1]$ is some region with the Lebesgue measure not greater than $2^{-K\beta p}$.

Proof It is shown in Theorem 3 of Schmidt-Hieber (2021) that there exists a ReLU-feedforward neural network function $f_{FF}^{(inner)}$ with depth $2K$ and width 4 such that

$$f_{FF}^{(inner)}(x) = \begin{cases} 0, & x \in (-\infty, 0); \\ \phi_K(x), & x \in [0, 1] \backslash \Omega^{(flaw)}; \\ 1, & x \in (1, \infty), \end{cases}$$

where $\Omega^{(flaw)} \subset [0, 1]$ is some region with the Lebesgue measure not greater than $2^{-K\beta p}$. Then the piecewise inner function $f_{FF}^{(R-pi)}$ can be constructed in the following way:

$$f_{FF}^{(R-pi)}(x) := \begin{pmatrix} 3 & 1 & \cdots & \frac{1}{3^{n-2}} \end{pmatrix} \begin{pmatrix} f_{FF}^{(inner)}(x) \\ f_{FF}^{(inner)}(x-2) \\ \vdots \\ f_{FF}^{(inner)}(x - 2(n-1)) \end{pmatrix} = \sum_{q=0}^{n-1} \frac{1}{3^{q-1}} f_{FF}^{(inner)}(x - 2q).$$

Table 2: Comparison of existing and our results on the approximation capabilities of different Transformer structures, where $\varepsilon \in (0,1)$ denotes the approximation error.

	Type	**Target Function**[1]	**Metric**	**Activations in Self-Attention Layers**[2]	**Activations in Feedforward Layers**	**Width**[3]	**Depth**
Yun et al. (2020a)	Universality	$\mathcal{C}^0$	L^p	σ_S with bias	σ_R		
Kajitsuka and Sato (2024)	Universality	$\mathcal{C}^0$	L^p	σ_S	σ_R		
Fang et al. (2022)	Universality	$\mathcal{C}^0$	L^∞	σ_H	σ_R		
Gurevych et al. (2022)	Rate	$\mathcal{F}(\mathcal{P})$	L^∞	$X \odot \sigma_H(X)$	σ_R	$\mathcal{O}(\max_{(p,K)\in\mathcal{P}} \varepsilon^{-K/p})$	$\mathcal{O}(1)$
Jiao et al. (2024)	Rate	$\mathcal{H}^\beta$ $\mathcal{C}^m$	L^∞	$X \odot \sigma_H(X)$	σ_R	$\mathcal{O}(\varepsilon^{-dn/\beta})$ $\mathcal{O}(\varepsilon^{-dn/m})$	$\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$
Takakura and Suzuki (2023)	Rate	$\mathcal{B}^{\gamma(\cdot;a)}_{p,\theta}$	L^2	σ_S	σ_R	$\mathcal{O}(\varepsilon^{-1/a^\dagger}\log\frac{1}{\varepsilon})$	$\mathcal{O}((\log\frac{1}{\varepsilon})^2)$
Havrilla and Liao (2024)	Rate	$\mathcal{H}^\beta$	L^∞	σ_R	σ_R	$\mathcal{O}(\varepsilon^{-dn/\beta})$	$\mathcal{O}(\log\frac{1}{\varepsilon})$
Jiao et al. (2025)	Rate	$\mathcal{H}^\beta$	L^∞	σ_S	σ_R	$\mathcal{O}(\varepsilon^{-dn/\beta})$	$\mathcal{O}(\log\frac{1}{\varepsilon})$
Ours (Theorem 6)	Rate	$\mathcal{H}^\beta$	L^∞	σ_S	$\sigma_R, \lfloor\cdot\rfloor$ $\sigma_R, \lfloor\cdot\rfloor, 2^x$ $\sigma_R, \lfloor\cdot\rfloor, \sigma_{NP}$ $\sigma_R, \lfloor\cdot\rfloor, \sigma_{RC}$	$\mathcal{O}(\varepsilon^{-\frac{2}{\beta}}\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(1)$ $\mathcal{O}(1)$	$\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$
Ours (Theorem 7)	Rate	$\mathcal{H}^\beta$	L^p	σ_S	$\sigma_R, sine, 2^x$ $\sigma_R, cosine, 3^x$ $\sigma_R, \sigma_P, \sigma_{NP}$ $\sigma_R, \sigma_P, \sigma_{RC}$	$\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(1)$ $\mathcal{O}(1)$	$\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\frac{1}{\varepsilon})$

1. $\mathcal{H}^\beta$ denotes the space of β-Hölder continuous functions with compact support and bounded norm. The space $\mathcal{C}^m$ consists of all functions whose first m derivatives exist and are continuous, and $\mathcal{C}^0$ denotes the space of continuous functions. $\mathcal{B}^{\gamma(\cdot;a)}_{p,\theta}$ refers to the Besov space with mixed and anisotropic smoothness defined in Takakura and Suzuki (2023). $\mathcal{F}(\mathcal{P})$ denotes the hierarchical composition model with smoothness and order constraint $\mathcal{P}$ defined in Gurevych et al. (2022). 2. We denote the softmax activation function by σ_S, the hardmax activation function by σ_H, and the ReLU activation function, which always operates component-wise, by σ_R. The symbol $\odot$ represents the Hadamard product. 3. Following Kim et al. (2023), the width of the Transformer network is defined as $\max\{hs, W\}$.

It is easy to verify that for $x \in [2q, 2q+1] \backslash (\Omega^{(flow)} + 2q), q \in \{0, 1, \dots, n-1\}$, there holds

$$f_{FF}^{(R-pi)}(x) = \frac{1}{3^{q-1}}\phi_K(x - 2q) + \frac{9}{2} - \frac{1}{2}\frac{1}{3^{q-2}}.$$

■

By incorporating the floor function as an activation function in the feedforward neural network, we can extend the above result to hold for all $x \in [0, 2n-1]$, and the required depth and width of the neural network function are reduced.

Lemma 16 *There exists a floor-ReLU-feedforward neural network function $f_{FF}^{(FR-pi)} : \mathbb{R} \to \mathbb{R}$ with depth $K+3$ and width $3n$ such that for $x \in [2q, 2q+1], q \in \{0, 1, \dots, n-1\}$, there holds*

$$f_{FF}^{(FR-pi)}(x) = \frac{1}{3^{q-1}}\phi_K(x - 2q).$$

Proof The proof is divided into two steps. In the first step, we show that $\phi_K(x)$ can be implemented by a floor-feedforward network function $f_{FF}^{(inner)}$(we call it as inner function) with depth $K+1$ and width 3 on $[0, 1]$ without any error. The second step is to adopt $f_{FF}^{(inner)}$ to construct the piecewise inner function $f_{FF}^{(FR-pi)}$.

Given $x = [0.a_1^x a_2^x \dots]_2 \in [0, 1]$, define $\{T_j\}_{j=1}^K$ to be

$$\begin{aligned} &T_1 = 2x; \\ &T_j = 2\left[\frac{1}{2^{K-j+1}}\sigma_F(2^{K-j+1}T_{j-1}) - \sigma_F(T_{j-1})\right], \quad j \in \{2, 3, \dots, K\}. \end{aligned} \tag{3}$$

It can be verified by the induction that, on the right-hand side in the definition of T_j for $j \geq 2$, the first term is actually T_{j-1} and the second term is the integer part of T_{j-1} and hence

$$T_j = [a_j^x.a_{j+1}^x \dots a_K^x]_2, \quad j \in \{2, 3, \dots, K\}.$$

We then define another sequence $\{S_j\}_{j=1}^K$:

$$\begin{aligned} &S_0 = 0; \\ &S_j = 2 \cdot 3^{dn(K-j)}\sigma_F(T_j) + \sigma_F(S_{j-1}), \quad j \in \{1, 2, \cdots, K\}. \end{aligned} \tag{4}$$

It is not hard to find that

$$S_j = 2\sum_{j'=1}^{j} a_{j'}^x 3^{dn(K-j')}.$$

Let

$$\begin{aligned} &f_0 = x; \\ &f_j = \begin{pmatrix} \sigma_F(2^{K-j}T_j) \\ \sigma_F(T_j) \\ \sigma_F(S_{j-1}) \end{pmatrix}, \quad j \in \{1, 2, \dots, K\}. \end{aligned}$$

$$\begin{tikzcd}
 & S_1 \arrow[r] & S_2 \arrow[r] & S_3 \arrow[r] & S_4 \arrow[r, "\times \frac{1}{3^{3dn+1}}"] & \phi_4(x) \\
x \arrow[r] & T_1 \arrow[u] \arrow[r] & T_2 \arrow[u] \arrow[r] & T_3 \arrow[u] \arrow[r] & T_4 \arrow[u] &
\end{tikzcd}$$

Figure 3: The flowchart for computing $\phi_K(x)$ when $K = 4$.

By (3)(4) we conclude that

$$\begin{aligned}
&f_0 = x; \\
&f_j = \sigma_F(A_j f_{j-1}), \quad j \in \{1, 2, \ldots, K\},
\end{aligned} \tag{5}$$

where

$$A_1 = \begin{pmatrix} 2^K \\ 2 \\ 0 \end{pmatrix}; \quad A_j = \begin{pmatrix} 1 & -2^{K-j+1} & 0 \\ \frac{1}{2^{K-j}} & -2 & 0 \\ 0 & 2 \cdot 3^{dn(K-j+1)} & 1 \end{pmatrix}, j \in \{2, 3, \ldots, K\}.$$

Define

$$f_{FF}^{(inner)} = A_{K+1} f_K \tag{6}$$

with

$$A_{K+1} = \frac{1}{3^{dn(K-1)+1}} \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}.$$

By (5)(6) we know that $f_{FF}^{(inner)}$ is a floor-feedforward network function with depth $K+1$ and width 3 and for any $x \in [0,1]$,

$$\begin{aligned}
f_{FF}^{(inner)} &= \frac{1}{3^{dn(K-1)+1}} [2\sigma_F(T_K) + \sigma_F(S_{K-1})] = \frac{1}{3^{dn(K-1)+1}} \left[2a_K^K + 2 \sum_{j=1}^{K-1} a_j^x 3^{dn(K-j)} \right] \\
&= \frac{1}{3^{dn(K-1)+1}} 2 \sum_{j=1}^{K} a_j^x 3^{dn(K-j)} = \phi_K(x).
\end{aligned}$$

Figure 3 illustrates the flowchart for computing $\phi_K(x)$ when $K = 4$.

Now we adpot $f_{FF}^{(inner)}$ to construct $f_{FF}^{(FR-pi)}$. Firstly, we use a ReLU-feedforward network to filter out $(-\infty, 0)$ and $(1, \infty)$, the intervals that we are not concerned with:

$$f_{FF}^{(filter)}(x) := -\sigma_R(-\sigma_R(x) + 1) + 1 = \begin{cases} 0, & x \in (-\infty, 0); \\ x, & x \in [0, 1]; \\ 1, & x \in (1, \infty). \end{cases}$$

The composition of $f_{FF}^{(inner)}$ and $f_{FF}^{(filter)}$ yields

$$f_{FF}^{(inner)}\left(f_{FF}^{(filter)}(x)\right) = \begin{cases} \phi_K(x), & x \in [0, 1]; \\ 0, & x \notin [0, 1]. \end{cases}$$

We can now construct the piecewise inner function $f_{FF}^{(FR-pi)}$ in the following way:

$$
\begin{aligned}
f_{FF}^{(FR-pi)}(x) := & \begin{pmatrix} 3 & 1 & \cdots & \frac{1}{3^{n-2}} \end{pmatrix} \begin{pmatrix} f_{FF}^{(inner)}\left(f_{FF}^{(filter)}(x)\right) \\ f_{FF}^{(inner)}\left(f_{FF}^{(filter)}(x-2)\right) \\ \vdots \\ f_{FF}^{(inner)}\left(f_{FF}^{(filter)}(x-2(n-1))\right) \end{pmatrix} \\
= & \sum_{q=0}^{n-1} \frac{1}{3^{q-1}} f_{FF}^{(inner)}\left(f_{FF}^{(filter)}(x-2q)\right).
\end{aligned}
$$

It is easy to verify that for $x \in [2q, 2q+1], q \in \{0, 1, \ldots, n-1\}$, there holds

$$
f_{FF}^{(FR-pi)}(x) = \frac{1}{3^{q-1}} \phi_K(x - 2q).
$$

∎

Appendix C. Memorization of Feedforward Neural Networks

In this section, we present some memorization results of feedforward neural networks. These results demonstrate the remarkable memorization capacity of feedforward neural networks: a very small neural network can memorize a vast dataset. To some extent, these findings theoretically reveal the reasons behind the tremendous success of neural networks in practical applications.

The memorization issue we are referring to here can be exact memorization or approximate memorization. That is, for a given set of data pairs $\{(x_m, y_m)\}_{m=1}^{M}$, the memory capability of feedforward neural networks implies the existence of a feedforward neural network function $f_{FF} : \mathbb{R} \to \mathbb{R}$ such that

$$
f_{FF}(x_m) = y_m, \quad m \in [M],
$$

where "=" can be replaced by "≈".

We first provide some results of exact binary memorization tasks.

Lemma 17 *Let $M \in \mathbb{N}_{\geq 1}$. For any $\{\theta_m\}_{m \in [M]} \subset \{0, 1\}^M$, there exists a feedforward neural network function $f_{FF}^{(mmr)} : \mathbb{R} \to \mathbb{R}$ such that*

$$
f_{FF}^{(mmr)}(m) = \theta_m, \quad \forall m \in [M].
$$

$f_{FF}^{(mmr)}$ can be in one of the following feedforward neural network classes:

(1) $\mathcal{FF}\left(7L-2, 2\left\lceil M^{1/L}\right\rceil + 2, \{ReLU, \text{ floor}\}\right)$, where L can be any positive integer number;

(2) $\mathcal{FF}(6, 3, \{ReLU, \text{ floor}, 2^x\})$;

(3) $\mathcal{FF}(4, 2, \{ReLU, \text{ sine}, 2^x\})$;

(4) $\mathcal{FF}(5, 1, \{ReLU, \text{ cosine}, 3^x\})$.

Proof (1) See Proposition 1 in Shen et al. (2021a).

(2) See Proposition 3.2 in Shen et al. (2021b). Note that σ_3 there is defined to be $\mathcal{T}(x - \lfloor x \rfloor - \frac{1}{2})$ with $\mathcal{T}(x)$ defined as $\mathbb{1}_{x\geq 0}$. Note that

$$\mathbb{1}_{x\geq 0} = \sigma_F(-\sigma_R(-\sigma_R(x+1)+1)+1),$$
$$x - \lfloor x \rfloor - \frac{1}{2} = \sigma_R(x) + \sigma_R(-x) - \sigma_F(x) - \frac{1}{2},$$

therefore σ_3 in Proposition 3.2 in Shen et al. (2021b) can be implemented by a ReLU-floor feedforward network function with depth 5 and width 3.

(3) See Lemma 3.1 in Jiao et al. (2023a).

(4) See Proposition 4.2 in Shen et al. (2021b). Note that ρ_3 there is defined to be $\widetilde{\mathcal{T}}(\cos(2\pi x))$ with $\widetilde{T}(x)$ defined as a piecewise linear function:

$$\widetilde{\mathcal{T}}(x) = \begin{cases} 0, & x \in (\cos(\frac{4\pi}{9}), \infty); \\ 1 - x/\cos(\frac{4\pi}{9}), & x \in [0, \cos(\frac{4\pi}{9})]; \\ 1, & x \in (-\infty, 0). \end{cases}$$

This piecewise linear function can be further reproduced by a ReLU feedforward network function with depth 3 and width 1:

$$\widetilde{\mathcal{T}}(x) = -\sigma_R\left(-\sigma_R\left(-\frac{x}{\cos\left(\frac{4\pi}{9}\right)} + 1\right) + 1\right) + 1.$$

■

Next, we introduce some results of memorization tasks where the labels can take on arbitrary values. These results make use of a well-known fact that an irrational winding on the torus is dense. Before we state this fact, we need to know the concept of rationally independent.

Definition 18 *For any $\{a_m\}_{m\in[M]} \subset \mathbb{R}$, they are called rationally independent if they are linearly independent over the rational numbers $\mathbb{Q}$. That is, if there exist $\lambda_1, \lambda_2, \cdots, \lambda_M \in \mathbb{Q}$ such that $\sum_{m=1}^{M} \lambda_m a_m = 0$, then $\lambda_1 = \lambda_2 = \cdots = \lambda_M = 0$.*

The following two lemmas provide two methods for generating rationally independent numbers.

Lemma 19 (Yarotsky (2021), Lemma 1) *Let $M \in \mathbb{N}_{\geq 1}$. There exists $w^{(1)}, w^{(0)} \in \mathbb{R}$ such that $\left\{\sigma_{NP}\left(w^{(1)}m + w^{(0)}\right)\right\}_{m\in[M]}$ are rationally independent.*

Lemma 20 (Zhang et al. (2022), Lemma 18) *Let $M \in \mathbb{N}_{\geq 1}$. $\{\sigma_{RC}(m)\}_{m\in[M]}$ are rationally independent.*

Remark 21 *Lemma 20 is a simplified version of Lemma 18 in Zhang et al. (2022), as the latter allows for replacing the sequence of positive integers $\{m\}_{m\in[M]}$ with any sequence of distinct rational numbers $\{r_m\}_{m\in[M]}$.*

The following proposition describes that an irrational winding on the torus is dense. For a proof, the readers are referred to Lemma 2 in Yarotsky (2021) and Lemma 19 in Zhang et al. (2022).

Proposition 22 *Let $M \in \mathbb{N}_{\geq 1}$. Given any rationally independent numbers $\{a_m\}_{m\in[M]}$ and an arbitrary periodic function $\sigma_P : \mathbb{R} \to \mathbb{R}$ with period T, i.e., $\sigma_P(x+T) = \sigma_P(x)$ for any $x \in \mathbb{R}$, assume there exist $x_1, x_2 \in \mathbb{R}$ with $0 < x_2 - x_1 < T$ such that σ_P is continuous on $[x_1, x_2]$. Then the following set*

$$\left\{ \left(\sigma_P\left(wa_1\right), \sigma_P\left(wa_2\right), \cdots, \sigma_P\left(wa_M\right)\right)^T : w \in \mathbb{R} \right\}$$

is dense in $\left[\sigma_P^{(\min)}, \sigma_P^{(\max)}\right]^M$, where $\sigma_P^{(\min)} := \min_{x\in[x_1,x_2]} \sigma_P(x)$ and $\sigma_P^{(\max)} := \max_{x\in[x_1,x_2]} \sigma_P(x)$. In the case of $\sigma_P^{(\min)} < \sigma_P^{(\max)}$, the following set

$$\left\{ \left(u \cdot \sigma_P\left(wa_1\right) + v, u \cdot \sigma_P\left(wa_2\right) + v, \cdots, u \cdot \sigma_P\left(wa_M\right) + v\right)^T : u, v, w \in \mathbb{R} \right\}$$

is dense in $\mathbb{R}^M$.

With these tools, we are able to construct feedforward neural networks that memorize the sequence of positive integers $\{m\}_{m\in[M]}$ to given real-valued labels. We can see that the networks in the following lemmas are fixed size and they can achieve any accuracy when memorizing integers.

Lemma 23 *Let $M \in \mathbb{N}_{\geq 1}$. For any $\epsilon > 0$ and any $\boldsymbol{\xi} \in \mathbb{R}^M$, there exist $w_r^{(0)}, w_r^{(1)}, w_r^{(2)}, w_r^{(3)}, w_r^{(4)} \in \mathbb{R}$ such that for $m \in [M]$, there holds*

$$\left| w_r^{(3)} \sigma_P\left(w_r^{(2)} \sigma_{NP}\left(w_r^{(0)} + w_r^{(1)} m \right)\right) + w_r^{(4)} - \xi_m \right| \leq \epsilon.$$

Proof A direct result from Lemma 19 and Proposition 22. ■

Lemma 24 *Let $M \in \mathbb{N}_{\geq 1}$. For any $\epsilon > 0$ and any $\boldsymbol{\xi} \in \mathbb{R}^M$, there exist $w^{(0)}, w^{(1)}, w^{(2)} \in \mathbb{R}$ such that for $m \in [M]$, there holds*

$$\left| w_r^{(1)} \sigma_P\left(w_r^{(0)} \sigma_{RC}\left(m\right)\right) + w_r^{(2)} - \xi_m \right| \leq \epsilon.$$

Proof A direct result from Lemma 20 and Proposition 22. ■

Appendix D. Proof of Lemma 12

We only present a proof for the ReLU case. The proof for the floor-ReLU case only differs subtly and we will point out these differences in the proof.

Let $f_{FF}^{(R-pi)}$ be the ReLU-feedforward network function in Lemma 15(For the floor-ReLU case, we make use of Lemma 16). Define the multi-dimensional piecewise inner function $\boldsymbol{f}_{FF}^{(mpi)} : \mathbb{R}^d \to \mathbb{R}^{2d}$ by

$$\begin{aligned}
\boldsymbol{f}_{FF}^{(mpi)}(\boldsymbol{x}) :&= \begin{pmatrix} \frac{1}{3}\mathbf{1}_{d\times1} & \frac{1}{3^{n+1}}\mathbf{1}_{d\times1} & \cdots & \frac{1}{3^{(d-1)n+1}}\mathbf{1}_{d\times1} \\ \mathbf{0}_{d\times1} & \mathbf{0}_{d\times1} & \cdots & \mathbf{0}_{d\times1} \end{pmatrix} \begin{pmatrix} f_{FF}^{(R-pi)}(x_1) \\ f_{FF}^{(R-pi)}(x_2) \\ \vdots \\ f_{FF}^{(R-pi)}(x_d) \end{pmatrix} \\
&= \begin{pmatrix} \left[\sum_{p=1}^d \frac{1}{3^{(p-1)n+1}} f_{FF}^{(R-pi)}(x_p)\right] \mathbf{1}_{d\times1} \\ \mathbf{0}_{d\times1} \end{pmatrix}.
\end{aligned}$$

By Lemma 15, it follows that for $\boldsymbol{x} \in \{[2q, 2q+1] \backslash (\Omega^{(flow)} + 2q)\}^d, q \in \{0, 1, \dots, n-1\}$, there holds

$$\boldsymbol{f}_{FF}^{(mpi)}(\boldsymbol{x}) = \begin{pmatrix} \left[\frac{1}{3^{q-1}} \sum_{p=1}^{d} \frac{1}{3^{(p-1)n+1}} \phi_K(x_p - 2q) + \frac{(3^q-1)(3^{dn}-1)}{2\cdot 3^{q-1}(3^{dn}-3^{(d-1)n})}\right] \mathbf{1}_{d\times 1} \\ \mathbf{0}_{d\times 1} \end{pmatrix}.$$

Let

$$\boldsymbol{b}_0 := \begin{pmatrix} 0\cdot \mathbf{1}_{d\times 1} & 2\cdot \mathbf{1}_{d\times 1} & \cdots & 2(n-1)\cdot \mathbf{1}_{d\times 1} \end{pmatrix} \in \mathbb{R}^{d\times n}.$$

Define $\boldsymbol{Z}_1 := \boldsymbol{\mathcal{F}}_{FF}^{(mpi)}(\boldsymbol{X} + \boldsymbol{b}_0) \in \mathbb{R}^{2d\times n}$. For $\boldsymbol{X}_{:,s} \in \{[0,1] \backslash \Omega^{(flow)}\}^d$, $\boldsymbol{X}_{:,s} + (\boldsymbol{b}_0)_{:,s} \in \{[2q, 2q+1] \backslash \Omega^{(flow)}\}^d$. Therefore it follows that

$$\boldsymbol{Z}_1 = \begin{pmatrix} \left[3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+1}} \phi_K(x_{p1} - 2q) + \frac{(3^0-1)(3^{dn}-1)}{2\cdot 3^{-1}(3^{dn}-3^{(d-1)n})}\right] \mathbf{1}_{1\times d} & \mathbf{0}_{1\times d} \\ \left[3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+2}} \phi_K(x_{p2} - 2q) + \frac{(3^1-1)(3^{dn}-1)}{2\cdot 3^{0-1}(3^{dn}-3^{(d-1)n})}\right] \mathbf{1}_{1\times d} & \mathbf{0}_{1\times d} \\ \vdots & \vdots \\ \left[3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+n}} \phi_K(x_{pn} - 2q) + \frac{(3^{n-1}-1)(3^{dn}-1)}{2\cdot 3^{n-2}(3^{dn}-3^{(d-1)n})}\right] \mathbf{1}_{1\times d} & \mathbf{0}_{1\times d} \end{pmatrix}^T.$$

Let

$$\boldsymbol{b}_1 := \begin{pmatrix} \frac{-(3^0-1)(3^{dn}-1)}{2\cdot 3^{-1}(3^{dn}-3^{(d-1)n})}\mathbf{1}_{d\times 1} & \frac{-(3^1-1)(3^{dn}-1)}{2\cdot 3^{0-1}(3^{dn}-3^{(d-1)n})}\mathbf{1}_{d\times 1} & \cdots & \frac{-(3^{n-1}-1)(3^{dn}-1)}{2\cdot 3^{n-2}(3^{dn}-3^{(d-1)n})}\mathbf{1}_{d\times 1} \\ \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times 1} & \cdots & \mathbf{0}_{d\times 1} \end{pmatrix} \in \mathbb{R}^{2d\times n}.$$

For the floor-ReLU case, we simply set $\boldsymbol{b}_1 = \mathbf{0}_{2d\times n}$. We then have

$$\boldsymbol{Z}_2 := \boldsymbol{Z}_1 + \boldsymbol{b}_1 =$$
$$\begin{pmatrix} 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+1}} \phi_K(x_{p1})\mathbf{1}_{d\times 1} & 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+2}} \phi_K(x_{p2})\mathbf{1}_{d\times 1} & \cdots & 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+n}} \phi_K(x_{pn})\mathbf{1}_{d\times 1} \\ \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times 1} & \cdots & \mathbf{0}_{d\times 1} \end{pmatrix} \in \mathbb{R}^{2d\times n}.$$

Next we define the matrices in the self-attention layer:

$$\boldsymbol{W}_O := \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_{d\times d} \\ \boldsymbol{I}_{d\times d} & \mathbf{0}_{d\times d} \end{pmatrix}, \quad \boldsymbol{W}_V := \boldsymbol{I}_{2d\times 2d}, \quad \boldsymbol{W}_K := \mathbf{0}_{2d\times 2d}, \quad \boldsymbol{W}_Q := \mathbf{0}_{2d\times 2d}.$$

Then

$$\boldsymbol{Z}_3 := \boldsymbol{\mathcal{F}}_{SA}(\boldsymbol{Z}_2) = \boldsymbol{Z}_2 + \boldsymbol{W}_O\boldsymbol{W}_V\boldsymbol{Z}_2\sigma_S((\boldsymbol{W}_V\boldsymbol{Z}_2)^T(\boldsymbol{W}_K\boldsymbol{Z}_2)) =$$
$$\begin{pmatrix} 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+1}} \phi_K(x_{p1})\mathbf{1}_{d\times 1} & 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+2}} \phi_K(x_{p2})\mathbf{1}_{d\times 1} & \cdots & 3\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+n}} \phi_K(x_{pn})\mathbf{1}_{d\times 1} \\ 3\sum_{q=1}^{n}\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+q}} \phi_K(x_{pq})\mathbf{1}_{d\times 1} & 3\sum_{q=1}^{n}\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+q}} \phi_K(x_{pq})\mathbf{1}_{d\times 1} & \cdots & 3\sum_{q=1}^{n}\sum_{p=1}^{d} \frac{1}{3^{(p-1)n+q}} \phi_K(x_{pq})\mathbf{1}_{d\times 1} \end{pmatrix}.$$

Define the scaling matrix

$$\boldsymbol{W}^{(scl)} := \begin{pmatrix} \mathbf{0}_{d\times d} & 3^{Kdn}\boldsymbol{I}_{d\times d} \end{pmatrix} \in \mathbb{R}^{d\times 2d}$$

and the segmentation matrix

$$\boldsymbol{b}^{(sgt)} := \begin{pmatrix} \mathbf{1}_{d\times 1} & (1+3^{Kdn})\mathbf{1}_{d\times 1} & \cdots & (1+(n-1)\cdot 3^{Kdn})\mathbf{1}_{d\times 1} \end{pmatrix} \in \mathbb{R}^{d\times n}.$$

Then $\boldsymbol{W}^{(scl)}\boldsymbol{Z}_3 + \boldsymbol{b}^{(sgt)}$ turns out to be the inner matrix $\boldsymbol{Z}$.

Appendix E. Proof of Lemma 13

Before we perform a binary expansion of g_{rs}, we need to normalize it so that its range lies in $[0,1]$. The normalization procedure is as follows:

$$g_{rs}^{(nml)}(x) := \frac{g_{rs}(x) - g_{\min}}{g_{\max} - g_{\min}}, \quad r \in [d], s \in [n].$$

Let

$$\Lambda := \left\{ 3^{Kdn+1} \sum_{q=1}^{n} \sum_{p=1}^{d} \frac{1}{3^{(p-1)n+q}} \phi_K(x_{pq}) + 1 + (s-1) \cdot 3^{Kdn} : \boldsymbol{x} \in [0,1]^{d \times n}, s \in [n] \right\}.$$

Now, given $r \in [n]$, for each

$$m = 3^{Kdn+1} \sum_{q=1}^{n} \sum_{p=1}^{d} \frac{1}{3^{(p-1)n+q}} \phi_K(x_{pq}) + 1 + (s-1) \cdot 3^{Kdn} \in \Lambda,$$

we perform a binary expansion of $g_{rs}^{(nml)}$ at the point $\frac{m-1-(s-1)\cdot 3^{Kdn}}{3^{Kdn}}$:

$$g_{rs}^{(nml)} \left(\frac{m - 1 - (s-1) \cdot 3^{Kdn}}{3^{Kdn}} \right) = \sum_{i=1}^{\infty} \theta_{rsi}^{(m)} 2^{-i},$$

where $\theta_{rsi}^{(m)} \in \{0,1\}$. For each $r \in [d]$ and $i \in [H]$, by Lemma 17, there exists a meomerization neural network function $f_{FF,ri}^{(mmr)}(x)$ such that

$$f_{FF,ri}^{(mmr)}(m) = \begin{cases} \theta_{rsi}^{(m)}, & m \in \Lambda; \\ 0, & \text{else.} \end{cases}$$

Here the number of points to be memorized is $M = n \cdot 3^{Kdn}$.

Arrange $\{f_{FF,ri}^{(mmr)}\}_{r\in[d], i\in[H]}$ in parallel to form a function from $\mathbb{R}^d$ to $\mathbb{R}^{dH}$:

$$\boldsymbol{f}_{FF}^{(mmr-prl)}(\boldsymbol{x}) :=$$
$$\left(f_{FF,11}^{(mmr)}(x_1) \quad \cdots \quad f_{FF,1H}^{(mmr)}(x_1) \quad f_{NN,21}^{(mmr)}(x_2) \quad \cdots \quad f_{FF,2H}^{(mmr)}(x_2) \quad \cdots \quad f_{FF,d1}^{(mmr)}(x_d) \quad \cdots \quad f_{FF,dH}^{(mmr)}(x_d) \right)^T.$$

The feedforward neural network function $\boldsymbol{f}_{FF}^{(pre-mmr)}$ that implements the approximate binary expansion is then defined by

$$\boldsymbol{f}_{FF}^{(pre-mmr)}(\boldsymbol{x}) := \boldsymbol{W}^{(coe)} \boldsymbol{f}_{FF}^{(mmr-prl)}(\boldsymbol{x}) = \sum_{i=1}^{H} 2^{-i} \begin{pmatrix} f_{FF,1i}^{(mmr)}(x_1) \\ f_{FF,2i}^{(mmr)}(x_2) \\ \vdots \\ f_{FF,di}^{(mmr)}(x_d) \end{pmatrix},$$

where the binary expansion coefficient matrix

$$\boldsymbol{W}^{(coe)} := \begin{pmatrix} \boldsymbol{w} & & & \\ & \boldsymbol{w} & & \\ & & \ddots & \\ & & & \boldsymbol{w} \end{pmatrix} \in \mathbb{R}^{d \times dH}$$

with

$$\boldsymbol{w} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \cdots & \frac{1}{2^H} \end{pmatrix} \in \mathbb{R}^{1\times H}.$$

The feedforward function $\boldsymbol{f}_{FF}^{(mmr)}(\boldsymbol{x})$ that we are looking for is now defined by

$$\boldsymbol{f}_{FF}^{(mmr)}(\boldsymbol{x}) := \boldsymbol{W}^{(nml)}\boldsymbol{f}_{FF}^{(pre-mmr)}(\boldsymbol{x}) + \boldsymbol{b}^{(nml)}$$

where

$$\boldsymbol{W}^{(nml)} := (g_{\max} - g_{\min})\boldsymbol{I}_{d\times d} \in \mathbb{R}^{d\times d}, \quad \boldsymbol{b}^{(nml)} := g_{\min}\mathbf{1}_{d\times n} \in \mathbb{R}^{d\times n}.$$

If follows that

$$\begin{aligned}
\widetilde{\boldsymbol{f}}_{:,s} &= \boldsymbol{W}^{(nml)}\boldsymbol{f}_{FF}^{(pre-mmr)}(\boldsymbol{Z}_{:,s}) + \boldsymbol{b}_{:,s}^{(nml)} \\
&= \begin{pmatrix} (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i} f_{FF,1i}^{(mmr)}(Z_{1s}) + g_{\min} \\ (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i} f_{FF,2i}^{(mmr)}(Z_{2s}) + g_{\min} \\ \vdots \\ (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i} f_{FF,di}^{(mmr)}(Z_{ds}) + g_{\min} \end{pmatrix} \\
&= \begin{pmatrix} (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i}\theta_{1si}^{(Z_{1s})} + g_{\min} \\ (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i}\theta_{2si}^{(Z_{2s})} + g_{\min} \\ \vdots \\ (g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i}\theta_{dsi}^{(Z_{ds})} + g_{\min} \end{pmatrix}.
\end{aligned}$$

Recall that

$$f_{rs}(\boldsymbol{X}) = g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)$$

and make use of the following equality:

$$\begin{aligned}
& g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right) \\
&= (g_{\max} - g_{\min})g_{rs}^{(nml)}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right) + g_{\min} \\
&= (g_{\max} - g_{\min})g_{rs}^{(nml)}\left(\frac{Z_{rs} - 1 - (s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right) + g_{\min} \\
&= (g_{\max} - g_{\min})\sum_{i=1}^{\infty} 2^{-i}\theta_{rsi}^{(Z_{rs})} + g_{\min},
\end{aligned}$$

we can now derive that

$$\begin{aligned}
& |\widetilde{f}_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})| \\
&= \left|(g_{\max} - g_{\min})\sum_{i=1}^{H} 2^{-i}\theta_{rsi}^{(Z_{rs})} + g_{\min} - \left((g_{\max} - g_{\min})\sum_{i=1}^{\infty} 2^{-i}\theta_{rsi}^{(Z_{rs})} + g_{\min}\right)\right.
\end{aligned}$$

$$
\begin{aligned}
&+g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right)-g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)\Bigg|\\
&\le (g_{\max}-g_{\min})\left|\sum_{i=1}^{H}2^{-i}\theta_{rsi}^{(Z_{rs})}-\sum_{i=1}^{\infty}2^{-i}\theta_{rsi}^{(Z_{rs})}\right|\\
&\quad+\left|g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right)-g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)\right|\\
&\le (g_{\max}-g_{\min})\sum_{i=H+1}^{\infty}2^{-i}\theta_{rsi}^{(Z_{rs})}+2^{\beta}Q\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}|\phi_K(x_{pq})-\phi(x_{pq})|\right)^{\log_3 2\cdot\frac{\beta}{dn}}\\
&\le \frac{g_{\max}-g_{\min}}{2^H}+\frac{2^{\beta}Q}{2^{(K+2)\beta}},
\end{aligned}
$$

where in the third step we use the fact $g_{rs}\in\mathcal{H}_{2^{\beta}Q}^{\frac{\beta\log 2}{dn\log 3}}(\mathcal{C})$ from Proposition 4 and in the fourth step we use the following estimate:

$$
|\phi_K(x)-\phi(x)|=2\sum_{j=K+1}^{\infty}a_j^x 3^{-dn(j-1)-1}\le\frac{2}{3^{2dn+1}-3^{dn+1}}\frac{1}{3^{dnK}}.
$$

Appendix F. Proof of Lemma 14

We only present a proof of (1). The proof of (2) is exactly the same. Let

$$
\Lambda:=\left\{3^{Kdn+1}\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})+1+(s-1)\cdot 3^{Kdn}:\boldsymbol{x}\in[0,1]^{d\times n},s\in[n]\right\}.
$$

Given $r\in[d]$, by Lemma 23 (for the case of (2), we apply Lemma 24), there exist $w_r^{(0)},w_r^{(1)},w_r^{(2)},w_r^{(3)},w_r^{(4)}$ such that for each

$$
m=3^{Kdn+1}\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})+1+(s-1)\cdot 3^{Kdn}\in\Lambda,
$$

there holds

$$
\left|w_r^{(3)}\sigma_P\left(w_r^{(2)}\sigma_{NP}\left(w_r^{(0)}+w_r^{(1)}m\right)\right)+w_r^{(4)}-g_{rs}\left(\frac{m-1-(s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right)\right|\le\delta.
$$

Define the memorization function

$$
f_{FF,r}^{(mmr)}(x):=w_r^{(3)}\sigma_P\left(w_r^{(2)}\sigma_{NP}\left(w_r^{(0)}+w_r^{(1)}x\right)\right)+w_r^{(4)},\quad r\in[d],
$$

then we have

$$
\left|f_{FF,r}^{(mmr)}(m)-g_{rs}\left(\frac{m-1-(s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right)\right|\le\delta.
$$

$\boldsymbol{f}_{FF}^{(mmr)}(\boldsymbol{x})$ is formed by arranging $\{f_{FF,r}^{(mmr)}\}_{r\in[d]}$ in parallel:

$$\boldsymbol{f}_{FF}^{(mmr)}(\boldsymbol{x}) := \begin{pmatrix} f_{FF,1}^{(mmr)}(x_1) & f_{FF,2}^{(mmr)}(x_2) & \cdots & f_{FF,d}^{(mmr)}(x_d) \end{pmatrix}^T .$$

It follows that

$$\widetilde{\boldsymbol{f}}_{:,s} = \boldsymbol{f}_{FF}^{(mmr)}(\boldsymbol{Z}_{:,s}) = \begin{pmatrix} f_{FF,1}^{(mmr)}(Z_{1s}) \\ f_{FF,2}^{(mmr)}(Z_{2s}) \\ \vdots \\ f_{FF,d}^{(mmr)}(Z_{ds}) \end{pmatrix} = \begin{pmatrix} w_1^{(3)}\sigma_P\left(w_1^{(2)}\sigma_{NP}\left(w_1^{(0)} + w_1^{(1)} Z_{1s}\right)\right) + w_1^{(4)} \\ w_2^{(3)}\sigma_P\left(w_2^{(2)}\sigma_{NP}\left(w_2^{(0)} + w_2^{(1)} Z_{1s}\right)\right) + w_2^{(4)} \\ \vdots \\ w_d^{(3)}\sigma_P\left(w_d^{(2)}\sigma_{NP}\left(w_d^{(0)} + w_d^{(1)} Z_{1s}\right)\right) + w_d^{(4)} \end{pmatrix}.$$

Recall that

$$f_{rs}\left(\boldsymbol{X}\right) = g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)$$

and make use of the following equality:

$$g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right) = g_{rs}\left(\frac{Z_{rs} - 1 - (s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right),$$

we can now derive that

$$\begin{aligned}
&|\widetilde{f}_{rs}(\boldsymbol{X}) - f_{rs}(\boldsymbol{X})| \\
&= \Bigg| w_r^{(3)}\sigma_P\left(w_r^{(2)}\sigma_{NP}\left(w_r^{(0)} + w_r^{(1)} Z_{rs}\right)\right) + w_r^{(4)} - g_{rs}\left(\frac{Z_{rs} - 1 - (s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right) \\
&\quad + g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right) - g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)\Bigg| \\
&\le \left| w_r^{(3)}\sigma_P\left(w_r^{(2)}\sigma_{NP}\left(w_r^{(0)} + w_r^{(1)} Z_{1s}\right)\right) + w_r^{(4)} - g_{rs}\left(\frac{Z_{rs} - 1 - (s-1)\cdot 3^{Kdn}}{3^{Kdn}}\right)\right| \\
&\quad + \left| g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi_K(x_{pq})\right) - g_{rs}\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}\phi(x_{pq})\right)\right| \\
&\le \delta + 2^{\beta} Q\left(3\sum_{q=1}^{n}\sum_{p=1}^{d}\frac{1}{3^{(p-1)n+q}}|\phi_K(x_{pq}) - \phi(x_{pq})|\right)^{\log_3 2\cdot\frac{\beta}{dn}} \\
&\le \delta + \frac{2^{\beta} Q}{2^{(K+2)\beta}},
\end{aligned}$$

where in the third step we use the fact $g_{rs} \in \mathcal{H}_{2^{\beta}Q}^{\frac{\beta\log 2}{dn\log 3}}(\mathcal{C})$ from Proposition 4 and in the fourth step we use the following estimate:

$$|\phi_K(x) - \phi(x)| = 2\sum_{j=K+1}^{\infty} a_j^x 3^{-dn(j-1)-1} \le \frac{2}{3^{2dn+1} - 3^{dn+1}}\frac{1}{3^{dnK}}.$$

References

Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in Neural Information Processing Systems*, 2021.

Richard Bellman. On the theory of dynamic programming. *Proceedings of the national Academy of Sciences*, 38(8):716–719, 1952.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*. PMLR.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and their ability to learn sparse boolean functions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*. Association for Computational Linguistics, 2023.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *International Conference on Machine Mearning*. PMLR, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in neural information processing systems*, 2021.

Xingwu Chen and Difan Zou. What can transformer learn with varying depth? case studies on sequence learning tasks. In *International Conference on Machine Learning*, 2024.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*. PMLR, 2021.

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

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

Zhiying Fang, Yidong Ouyang, Ding-Xuan Zhou, and Guang Cheng. Attention enables zero approximation error. *arXiv preprint arXiv:2202.12166*, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.

Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

Ingo Gühring, Gitta Kutyniok, and Philipp Petersen. Error bounds for approximations with deep relu neural networks in w s, p norms. *Analysis and Applications*, 18(05):803–859, 2020.

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

Alexander 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.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Jerry Yao-Chieh Hu, Wei-Po Wang, Ammar Gilani, Chenyang Li, Zhao Song, and Han Liu. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. In *The Thirteenth International Conference on Learning Representations, Singapore, April 24-28*, 2025.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in neural information processing systems*, 2021.

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

Yuling Jiao, Yanming Lai, Xiliang Lu, Fengru Wang, Jerry Zhijian Yang, and Yuanyuan Yang. Deep neural networks with relu-sine-exponential activations break curse of dimensionality in approximation on hölder class. *SIAM Journal on Mathematical Analysis*, 55(4):3635–3649, 2023a.

Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, 2023b.

Yuling Jiao, Yanming Lai, Yang Wang, and Bokai Yan. Convergence analysis of flow matching in latent space with transformers. *arXiv preprint arXiv:2404.02538*, 2024.

Yuling Jiao, Yanming Lai, Defeng Sun, Yang Wang, and Bokai Yan. Approximation bounds for transformed with application to regression. *preprint*, 2025.

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

Tokio Kajitsuka and Issei Sato. On the optimal memorization capacity of transformers. In *The Thirteenth International Conference on Learning Representations, Singapore, April 24-28*, 2025.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*. PMLR, 2021.

Junghwan Kim, Michelle Kim, and Barzan Mozafari. Provable memorization capacity of transformers. In *International Conference on Learning Representations*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ker Ko. *Complexity theory of real functions*. Springer Science & Business Media, 2012.

Andrei Nikolaevich Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*, volume 114, pages 953–956. Russian Academy of Sciences, 1957.

Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanic. Universal approximation under constraints is possible with transformers. In *International Conference on Learning Representations*, 2022.

Valerii Likhosherstov, Krzysztof Choromanski, and Adrian Weller. On the expressive flexibility of self-attention matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, 2017.

Shengjie Luo, Shanda Li, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. Your transformer may not be as powerful as you expect. In *Advances in Neural Information Processing Systems*, 2022.

Liam Madden, Curtis Fox, and Christos Thrampoulidis. Upper and lower memory capacity bounds of transformers for next-token prediction. *arXiv preprint arXiv:2405.13718*, 2024.

Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. Memorization capacity of multi-head attention in transformers. In *International Conference on Learning Representations*, 2024.

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

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision*, 2023.

Aleksandar Petrov, Philip HS Torr, and Adel Bibi. Prompting a pretrained transformer can be a universal approximator. In *Proceedings of the 41st International Conference on Machine Learning*, pages 40523–40550, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya. Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. In *Advances in Neural Information Processing Systems*, 2024.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875, 2020.

Johannes Schmidt-Hieber. The kolmogorov–arnold representation theorem revisited. *Neural networks*, 137:119–126, 2021.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28(5):1768–1811, 2020.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network with approximation error being reciprocal of width to power of square root of depth. *Neural Computation*, 33(4):1005–1036, 2021a.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Neural network approximation: Three hidden layers are enough. *Neural Networks*, 141:160–173, 2021b.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Optimal approximation rate of relu networks in terms of width and depth. *Journal de Mathématiques Pures et Appliquées*, 157:101–135, 2022.

Jonathan W Siegel. Optimal approximation rates for deep relu neural networks on sobolev and besov spaces. *Journal of Machine Learning Research*, 24(357):1–52, 2023.

Jonathan W Siegel and Jinchao Xu. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks. *Foundations of Computational Mathematics*, 24(2):481–537, 2024.

Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

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

Jacob Trauger and Ambuj Tewari. Sequence length independent norm-based generalization bounds for transformers. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024.

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

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

Mingze Wang and Weinan E. Understanding the expressive power and mechanisms of transformer for sequence modeling. In *Advances in Neural Information Processing Systems*, 2024.

Qianchao Wang, Shijun Zhang, Dong Zeng, Zhaoheng Xie, Hengtao Guo, Tieyong Zeng, and Feng-Lei Fan. Don't fear peculiar activation functions: Euaf and beyond. *Neural Networks*, 186: 107258, 2025.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D. Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In *Forty-first International Conference on Machine Learning, Vienna, Austria, July 21-27*, 2024.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. In *Advances in Neural Information Processing Systems*, 2022.

Noam Wies, Yoav Levine, Daniel Jannai, and Amnon Shashua. Which transformer architecture fits my data? a vocabulary bottleneck in self-attention. In *International Conference on Machine Learning*. PMLR, 2021.

Yunfei Yang. On the optimal approximation of sobolev and besov functions using deep relu neural networks. *Applied and Computational Harmonic Analysis*, page 101797, 2025.

Yunfei Yang and Ding-Xuan Zhou. Nonparametric regression using over-parameterized shallow relu neural networks. *Journal of Machine Learning Research*, 25(165):1–35, 2024a.

Yunfei Yang and Ding-Xuan Zhou. Optimal rates of approximation by shallow relu k neural networks and applications to nonparametric regression. *Constructive Approximation*, pages 1–32, 2024b.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, 2019.

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

Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. In *Conference on Learning Theory*. PMLR, 2018.

Dmitry Yarotsky. Elementary superexpressive activations. In *International Conference on Machine Learning*. PMLR, 2021.

Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, 2021.

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

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

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

Shijun Zhang, Zuowei Shen, and Haizhao Yang. Deep network approximation: Achieving arbitrary accuracy with fixed number of neurons. *Journal of Machine Learning Research*, 23(276):1–60, 2022.