Vocabulary In-Context Learning in Transformers: Benefits of Positional Encoding

Anonymous Author(s)
Affiliation
Address
email

Abstract

Numerous studies have demonstrated that the Transformer architecture possesses the capability for in-context learning (ICL). In scenarios involving function approximation, context can serve as a control parameter for the model, endowing it with the universal approximation property (UAP). In practice, context is represented by tokens from a finite set, referred to as a vocabulary, which is the case considered in this paper, *i.e.,* vocabulary in-context learning (VICL). We demonstrate that VICL in single-layer Transformers, without positional encoding, does not possess the UAP; however, it is possible to achieve the UAP when positional encoding is included. Several sufficient conditions for the positional encoding are provided. Our findings reveal the benefits of positional encoding from an approximation theory perspective in the context of ICL.

1 Intruduction

Transformers have emerged as a dominant architecture in deep learning over the past few years. Thanks to their remarkable performance in language tasks, they have become the preferred framework in the natural language processing (NLP) field. A major trend in modern NLP is the development and integration of various black-box models, along with the construction of extensive text datasets. In addition, improving model performance in specific tasks through techniques such as in-context learning (ICL) [1, 2], chain of thought (CoT) [3, 4], and retrieval-augmented generation (RAG) [5] has become a significant research focus. While the practical success of these models and techniques is well-documented, the theoretical understanding of why they perform so well remains incomplete.

To explore the capabilities of Transformers in handling ICL tasks, it is essential to examine their approximation power. The universal approximation property (UAP) [6–9] has long been a key topic in the theoretical study of neural networks (NNs), with much of the focus historically on feed-forward neural networks (FNNs). Yun et al. [10] was the first to investigate the UAP of Transformers, demonstrating that any sequence-to-sequence function could be approximated by a Transformer network with fixed positional encoding. Luo et al. [11] highlighted that a Transformer with relative positional encoding does not possess the UAP. Meanwhile, Petrov et al. [12] explored the role of prompting in Transformers, proving that prompting a pre-trained Transformer can act as a universal functional approximator.

However, one limitation of these studies is that, in practical scenarios, the inputs to language models are derived from a finite set embedded in high-dimensional Euclidean space – commonly referred to as a vocabulary. Whether examining the work on prompts in [12] or the research on ICL in [13, 14], these studies assume inputs from the entire Euclidean space, which differs significantly from the discrete nature of vocabularies used in real-world applications.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

1.1 Contributions

Starting with the connection between FNNs and Transformers, we turn to the finite restriction of vocabularies and study the benefits of positional encoding. Leveraging the UAP of FNNs, we explore the approximation properties of Transformers for ICL tasks in two scenarios: one where positional encoding is used and one where it is not. In both cases, the inputs are from a finite vocabulary. More specifically:

1. We first establish a connection between FNNs and Transformers in processing ICL tasks (Lemma 3). Using this lemma, we show that Transformers can function as universal approximators (Lemma 4), where the context serves as control parameters, while the weights and biases of the Transformer remain fixed.
2. When the vocabulary is finite and positional encoding is not used, we prove that single-layer Transformers cannot achieve the UAP for ICL tasks (Theorem 7).
3. However, when positional encoding is used, it becomes possible for single-layer Transformers to achieve the UAP (Theorem 8). In particular, for Transformers with ReLU or softmax activation functions, the conditions on the positional encoding are relaxed (Theorem 9).

1.2 Related Works

Universal approximation property. NNs, through multi-layer nonlinear transformations and feature extraction, are capable of learning deep feature representations from raw data. As neural networks gain prominence, theoretical investigations—especially into their UAP – have intensified. Related studies typically fall into two categories: those allowing arbitrary width with fixed depth [6–9], and those allowing arbitrary depth with bounded width [15–18]. Since our study builds on existing results regarding the approximation capabilities of FNNs, we focus on investigating the approximation abilities of single-layer Transformers in modulating context for ICL tasks. Consequently, our work relies more on the findings from the first category of research. The realization of the UAP depends on the architecture of the network itself, providing constructive insights for exploring the connection between FNNs and Transformers. Recently, Petrov et al. [12] also explored UAP in the context of ICL, but without considering vocabulary constraints or positional encodings.

Transformers. The Transformer is a widely used neural network architecture for modeling sequences [19–24]. This non-recurrent architecture relies entirely on the attention mechanism to capture global dependencies between inputs and outputs [19]. The highly effective neural sequence transduction model is typically structured using an encoder-decoder framework [25, 26].

Without positional encoding, the Transformer can be viewed as a stack of N blocks, each consisting of a self-attention layer followed by a feed-forward layer with skip connections. In this paper, we focus on the case of a single-layer self-attention sequence encoder.

In-context learning. The Transformer has demonstrated remarkable performance in the field of NLP, and large language models (LLMs) are gaining increasing popularity. ICL has emerged as a new paradigm in NLP, enabling LLMs to make better predictions through prompts provided within the context [2, 27–30]. ICL delivers high performance with high-quality data at a lower cost [31–33]. It enhances retrieval-augmented methods by prepending grounding documents to the input [34] and can effectively update or refine the model's knowledge base through well-designed prompts [35].

Positional Encoding. The following explanation clarifies the significance of incorporating positional encoding into the Transformer architecture. RNNs capture sequential order by encoding the changes in hidden states over time. In contrast, for Transformers, the self-attention mechanism is permutation equivariant, meaning that for any model f, any permutation matrix π, and any input x, the following holds: $f(\pi(x)) = \pi(f(x))$.

We aim to explore the impact of positional encoding on the performance of a single-layer Transformer when performing ICL tasks with a finite vocabulary. Therefore, we focus on analyzing existing positional encoding methods. There are fundamental methods for encoding positional information in a sequence within the Transformer: absolute positional encodings (APEs) *e.g.* [36, 24, 37, 38], relative positional encodings (RPEs) *e.g.* [39, 40, 38] and rotary positional embedding (RoPE) [41].

The commonly used APE is implemented by directly adding the positional encodings to the word embeddings, and we follow this implementation.

UAP of ICL. Regarding the understanding of the mechanism of ICL, various explanations have been proposed, including those based on Bayesian theory [42, 43] and gradient descent theory [44]. Fine-tuning the Transformer through ICL alters the presentation of the input rather than the model parameters, which is driven by successful few-shot and zero-shot learning [45, 46]. This success raises the question of whether we can achieve the UAP through context adjustment.

Yun et al. [10] demonstrated that Transformers can serve as universal sequence-to-sequence approximators, while Alberti et al. [47] extended the UAP to architectures with non-standard attention mechanisms. However, their implementations allow the internal parameters of the Transformers to vary, which does not fully reflect the characteristics of ICL. In contrast, Likhosherstov et al. [48] showed that while the parameters of self-attention remain fixed, various sparse matrices can be approximated by altering the inputs. Fixing self-attention parameters aligns more closely with practical scenarios and provides valuable insights for our work. However, this approach has the limitation of excluding the full Transformer architecture. Furthermore, Deora et al. [49] illustrated the convergence and generalization of single-layer multi-head self-attention models trained using gradient descent, supporting the feasibility of our research by emphasizing the robust generalization of Transformers. Nevertheless, Petrov et al. [50] indicated that the presence of a prefix does not alter the attention focus within the context, prompting us to explore variations in input context and introduce flexibility in positional encoding.

1.3 Outline

We will introduce the notations and background results in Section 2. Section 3 addresses the case where the vocabulary is finite and positional encoding is not used. Section 4 discusses the benefits of using positional encoding. A summary is provided in Section 5. All proof of lemmas and theorems are provided in appendix.

2 Background Materials

We consider the approximation problem as follows. Given a fixed Transformer network, for any target continuous function $f : \mathcal{K} \to \mathbb{R}^{d_y}$ with a compact domain $\mathcal{K} \subset \mathbb{R}^{d_x}$, we aim to adjust the content of the context so that the output of the Transformer network can approximate f. First, we present the concrete forms and notations for the inputs of ICL, FNNs, and Transformers.

2.1 Notations

Input of in-context learning. In the ICL task, the given n demonstrations are denoted as $z^{(i)} = (x^{(i)}, y^{(i)})$ for $i = 1, 2, ..., n$, where $x^{(i)} \in \mathbb{R}^{d_x}$ and $y^{(i)} \in \mathbb{R}^{d_y}$. Unlike the setting in [13, 14] where $y^{(i)}$ was related to $x^{(i)}$ (for example $y^{(i)} = \phi(x^{(i)})$ for some function ϕ), we do not assume any correspondence between $x^{(i)}$ and $y^{(i)}$, *i.e.* , $x^{(i)}$ and $y^{(i)}$ are chosen freely. To predict the target at a query vector $x \in \mathbb{R}^{d_x}$ or $z = (x, 0) \in \mathbb{R}^{d_x+d_y}$, we define the input matrix Z as following:

$$Z = \begin{bmatrix} z^{(1)} & z^{(2)} & \cdots & z^{(n)} & z \end{bmatrix} := \begin{bmatrix} x^{(1)} & x^{(2)} & \cdots & x^{(n)} & x \\ y^{(1)} & y^{(2)} & \cdots & y^{(n)} & 0 \end{bmatrix} \in \mathbb{R}^{(d_x+d_y)\times(n+1)}. \quad (1)$$

Furthermore, let $\mathcal{P} : \mathbb{N}^+ \to \mathbb{R}^{d_x+d_y}$ represent a positional encoding function, and define $\mathcal{P}^{(i)} := \mathcal{P}(i)$. Denote the demonstrations with positional encoding as $z_{\mathcal{P}}^{(i)} := z^{(i)} + \mathcal{P}^{(i)}$ and $z_{\mathcal{P}} := z + \mathcal{P}^{(n+1)}$. The context with positional encoding can then be represented as:

$$Z_{\mathcal{P}} = \begin{bmatrix} z_{\mathcal{P}}^{(1)} & z_{\mathcal{P}}^{(2)} & \cdots & z_{\mathcal{P}}^{(n)} & z_{\mathcal{P}} \end{bmatrix} := \begin{bmatrix} x_{\mathcal{P}}^{(1)} & x_{\mathcal{P}}^{(2)} & \cdots & x_{\mathcal{P}}^{(n)} & x_{\mathcal{P}} \\ y_{\mathcal{P}}^{(1)} & y_{\mathcal{P}}^{(2)} & \cdots & y_{\mathcal{P}}^{(n)} & y_{\mathcal{P}} \end{bmatrix} \in \mathbb{R}^{(d_x+d_y)\times(n+1)}. \quad (2)$$

Additionally, we denote:

$$X = \begin{bmatrix} x^{(1)} & x^{(2)} & \cdots & x^{(n)} \end{bmatrix} \in \mathbb{R}^{d_x \times n}, \qquad X_{\mathcal{P}} = \begin{bmatrix} x_{\mathcal{P}}^{(1)} & x_{\mathcal{P}}^{(2)} & \cdots & x_{\mathcal{P}}^{(n)} \end{bmatrix} \in \mathbb{R}^{d_x \times n}, \quad (3)$$

$$Y = \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(n)} \end{bmatrix} \in \mathbb{R}^{d_y \times n}, \qquad Y_{\mathcal{P}} = \begin{bmatrix} y_{\mathcal{P}}^{(1)} & y_{\mathcal{P}}^{(2)} & \cdots & y_{\mathcal{P}}^{(n)} \end{bmatrix} \in \mathbb{R}^{d_y \times n}. \quad (4)$$

Feed-forward neural networks. One-hidden-layer FNNs have sufficient capacity to approximate continuous functions on any compact domain. In this article, all the FNNs we refer to and use are one-hidden-layer networks. We denote a one-hidden-layer FNN with activation function σ as N^σ, and the set of all such networks is denoted as $\mathcal{N}^\sigma$, *i.e.*,

$$\begin{aligned}\mathcal{N}^\sigma &= \left\{\mathrm{N}^\sigma := A\,\sigma(Wx+b) \;\middle|\; A \in \mathbb{R}^{d_y\times k},\ W\in\mathbb{R}^{k\times d_x},\ b\in\mathbb{R}^k,\ k\in\mathbb{N}^+\right\} \\ &= \left\{\mathrm{N}^\sigma := \sum_{i=1}^k a_i\sigma(w_i\cdot x+b_i) \;\middle|\; (a_i,w_i,b_i)\in\mathbb{R}^{d_y}\times\mathbb{R}^{d_x}\times\mathbb{R},\ k\in\mathbb{N}^+\right\}.\end{aligned} \tag{5}$$

For element-wise activations, such as ReLU, the above notation is well-defined. However, for not widely used but considered in this article non element-wise activation function, especially softmax activation, we need to give more details for the notation:

$$\mathcal{N}^{\mathrm{softmax}} = \left\{\mathrm{N}^{\mathrm{softmax}} = \frac{\sum_{i=1}^k a_i \mathrm{e}^{w_i\cdot x+b_i}}{\sum_{i=1}^k \mathrm{e}^{w_i\cdot x+b_i}} \;\middle|\; (a_i,w_i,b_i)\in\mathbb{R}^{d_y}\times\mathbb{R}^{d_x}\times\mathbb{R},\ k\in\mathbb{N}^+\right\}. \tag{6}$$

Transformers. We define the general attention mechanism following [13, 14] as:

$$\mathrm{Attn}^\sigma_{Q,K,V}(Z) := VZM\sigma\big((QZ)^\top KZ\big), \tag{7}$$

where $V,\ Q,\ K$ are the value, query, and key matrices in $\mathbb{R}^{(d_x+d_y)\times(d_x+d_y)}$, respectively. $M = \mathrm{diag}(I_n, 0)$ is the mask matrix in $\mathbb{R}^{(n+1)\times(n+1)}$, and σ is the activation function. Here the softmax activation of a matrix $G\in\mathbb{R}^{m\times n}$ is defined as:

$$\big(\mathrm{softmax}(G)\big)_{i,j} := \frac{\exp\big(G_{i,j}\big)}{\sum_{l=1}^m \exp\big(G_{l,j}\big)}. \tag{8}$$

With this formulation of the general attention mechanism, we can define a single-layer Transformer without positional encoding as:

$$\mathrm{T}^\sigma(x;X,Y) := (Z + VZM\sigma((QZ)^\top KZ))_{d_x+1:d_x+d_y,n+1}, \tag{9}$$

where $[a:b, c:d]$ denotes the submatrix from the a-th row to the b-th row and from the c-th column to the d-th column. If $a=b$ (or $c=d$), the row (or column) index is reduced to a single number. Similarly to the notation for FNNs, $\mathcal{T}^\sigma$ denotes the set of all T^σ with different parameters.

Vocabulary. In the above notations, the tokens in context of ICL are general and unrestricted. When we refer to a "vocabulary", we mean that the tokens are drawn from a finite set. More specifically, we refer to it as VICL if all input vectors $z^{(i)}$ come from a finite vocabulary $\mathcal{V} = \mathcal{V}_x\times\mathcal{V}_y \subset \mathbb{R}^{d_x}\times\mathbb{R}^{d_y}$. In this case, we use subscript $*$, *i.e.* $\mathrm{T}^\sigma_*(x;X,Y)$, to represent the Transformer $\mathrm{T}^\sigma(x;X,Y)$ defined in equation (9), and denote the set of such Transformers as $\mathcal{T}^\sigma_*$:

$$\mathcal{T}^\sigma_* = \left\{\mathrm{T}^\sigma_*(x;X,Y) := \mathrm{T}^\sigma(x;X,Y) \;\middle|\; z^{(i)}\in\mathcal{V},\ i\in\{1,2,\cdots,n\},\ n\in\mathbb{N}^+\right\}. \tag{10}$$

To facilitate the simplification of VICL analysis, we denote a FNN with a finite set of weights as N^σ_*, and the corresponding set of such networks as $\mathcal{N}^\sigma_*$. More specifically, when the activation function is an elementwise activation, we denote:

$$\mathcal{N}^\sigma_* = \left\{\mathrm{N}^\sigma_* := \sum_{i=1}^k a_i\sigma(w_i\cdot x+b_i) \;\middle|\; (a_i,w_i,b_i)\in\mathcal{A}\times\mathcal{W}\times\mathcal{B},\ k\in\mathbb{N}^+\right\}. \tag{11}$$

where $\mathcal{A}\subset\mathbb{R}^{d_y},\ \mathcal{W}\subset\mathbb{R}^{d_x}$, and $\mathcal{B}\subset\mathbb{R}$ are finite sets. When the activation function is $\mathrm{softmax}$, we denote:

$$\mathcal{N}^{\mathrm{softmax}} = \left\{\mathrm{N}^{\mathrm{softmax}} = \frac{\sum_{i=1}^k a_i e^{w_i\cdot x+b_i}}{\sum_{i=1}^k e^{w_i\cdot x+b_i}} \;\middle|\; (a_i,w_i,b_i)\in\mathcal{A}\times\mathcal{W}\times\mathcal{B},\ k\in\mathbb{N}^+\right\} \tag{12}$$

where $\mathcal{A},\mathcal{W}$ and $\mathcal{B}$ are defined as in the previous context.

Positional encoding. When positional encoding $\mathcal{P}$ is involved, we add the subscript $\mathcal{P}$, *i.e.*,

$$\mathcal{T}^{\sigma}_{*,\mathcal{P}} = \left\{ \mathrm{T}^{\sigma}_{*,\mathcal{P}}(x; X, Y) := \mathrm{T}^{\sigma}(x_{\mathcal{P}}; X_{\mathcal{P}}, Y_{\mathcal{P}}) \;\middle|\; z^{(i)} \in \mathcal{V}, i \in \{1, 2, ..., n\}, n \in \mathbb{N}^{+} \right\}. \tag{13}$$

Note that the context length n in T^{σ}, T^{σ}_{*} and $\mathrm{T}^{\sigma}_{*,\mathcal{P}}$ are unbounded.

We present all our notations in Table 1 in Appendix A for easy reference.

2.2 Universal Approximation Property

The vanilla form of the UAP for FFNs plays a crucial role in our study. Before we state this property as a formal lemma, we put forward the following assumption first, which is similar to the one in [14] and is used to simplify the analysis of Transformers.

Assumption 1. *The matrices $Q,\ K,\ V \in \mathbb{R}^{(d_x+d_y)\times(d_x+d_y)}$ have the following sparse partition:*

$$Q = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} D & E \\ F & U \end{bmatrix}, \tag{14}$$

where $B,\ C,\ D \in \mathbb{R}^{d_x\times d_x}$, $E \in \mathbb{R}^{d_x\times d_y}$, $F \in \mathbb{R}^{d_y\times d_x}$ and $U \in \mathbb{R}^{d_y\times d_y}$. Furthermore, the matrices $B,\ C$ and U are non-singular, and the matrix $F = 0$.

In addition, we assume the element-wise activation σ is non-polynomial, locally bounded, and continuous. In fact, this assumption can be weakened, which will be discussed in Appendix F. Here, we have slightly strengthened it for the sake of computational convenience.

Lemma 2 (UAP of FNNs [9]). *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-polynomial, locally bounded, piecewise continuous activation function. For any continuous function $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$, and for any $\varepsilon > 0$, there exist $k \in \mathbb{N}^{+}$, $A \in \mathbb{R}^{d_y\times k}$, $b \in \mathbb{R}^{k}$, and $W \in \mathbb{R}^{k\times d_x}$ such that*

$$\|A\sigma(Wx + b) - f(x)\| < \varepsilon, \quad \forall x \in \mathcal{K}. \tag{15}$$

The theorem presented above is well-known and primarily applies to activation functions operating element-wise. However, it can be readily extended to the case of the softmax activation function. In fact, this can be achieved using NNs with exponential activation functions. The specific approach for this generalization is detailed in Appendix B.

2.3 Feed-forward neural networks and Transformers

It is important to emphasize the connection between FNNs and Transformers, which will be represented in the following lemmas and are crucial for establishing our main theory.

Lemma 3. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-polynomial, locally bounded, piecewise continuous activation function, and T^{σ} be a single-layer Transformer satisfying Assumption 1. For any one-hidden-layer network $\mathrm{N}^{\sigma} : \mathbb{R}^{d_x-1} \to \mathbb{R}^{d_y} \in \mathcal{N}^{\sigma}$ with n hidden neurons, there exist matrices $X \in \mathbb{R}^{d_x\times n}$ and $Y \in \mathbb{R}^{d_y\times n}$ such that*

$$\mathrm{T}^{\sigma}\left(\tilde{x}; X, Y\right) = \mathrm{N}^{\sigma}(x), \quad \forall x \in \mathbb{R}^{d_x-1}. \tag{16}$$

There is a difference in the input dimensions of T^{σ} and N^{σ}, as the latter includes a bias dimension absent in the former. To connect the two inputs, $\tilde{x}$ and x, we use a tilde, where $\tilde{x}$ is formed by augmenting x with an additional one appended to the end.

By employing the structure of K, Q and V in equation (14), the output forms of the Transformer $\mathrm{T}^{\sigma}\big(\tilde{x}; X, Y\big)$ can be simplified as follows:

$$\begin{aligned} \mathrm{T}^{\sigma}\big(\tilde{x}; X, Y\big) &= \left(\begin{bmatrix} X & \tilde{x} \\ Y & 0 \end{bmatrix} + \begin{bmatrix} DX + EY & 0 \\ FX + UY & 0 \end{bmatrix} \sigma\left(\begin{bmatrix} X^{\top}B^{\top}CX & X^{\top}B^{\top}C\tilde{x} \\ \tilde{x}^{\top}B^{\top}CX & \tilde{x}^{\top}B^{\top}C\tilde{x} \end{bmatrix} \right) \right)_{d_x+1:d_x+d_y,\, n+1} \\ &= (FX + UY)\sigma(X^{\top}B^{\top}C\tilde{x}) = UY\sigma(X^{\top}B^{\top}C\tilde{x}). \end{aligned} \tag{17}$$

Comparing this with the output form of FNNs, *i.e.*, $\mathrm{N}^{\sigma}(x) = A\sigma(Wx + b)$, it becomes evident that setting $X = (C^{\top}B)^{-1}\begin{bmatrix} W & b \end{bmatrix}^{\top}$ and $Y = U^{-1}A$ is sufficient to finish the proof.

It can be observed that the form in equation (17) exhibits the structure of an FNN. Consequently, Lemma 3 implies that single-layer Transformers T^σ in the context of ICL and FNNs N^σ are equivalent. However, this equivalence does not hold for the case of softmax activation due to differences in the normalization operations between FNNs and Transformers. Therefore, in the subsequent sections of this article, we employ different analytical methods to address the two types of activation functions.

Moreover, the equivalence in equation (16) suggests that the context in Transformers can act as a control parameter for the model, thereby endowing it with the UAP.

2.4 Universal Approximation Property of In-context Learning

We now present the UAP of Transformers in the context of ICL.

Lemma 4. *Let σ be a non-polynomial, locally bounded, piecewise continuous activation function or softmax activation function, and T^σ be a single-layer Transformer satisfying Assumption 1, and $\mathcal{K}$ be a compact domain in $\mathbb{R}^{d_x-1}$. Then for any continuous function $f:\mathcal{K}\to\mathbb{R}^{d_y}$ and any $\varepsilon>0$, there exist matrices $X\in\mathbb{R}^{d_x\times n}$ and $Y\in\mathbb{R}^{d_y\times n}$ such that*

$$\left\|\mathrm{T}^\sigma\big(\tilde{x};X,Y\big)-f(x)\right\|<\varepsilon,\quad \forall x\in\mathcal{K}. \tag{18}$$

For the case of element-wise activation, the result follows directly by combining Lemma 2 and Lemma 3. However, for the softmax activation, the normalization operation requires an additional technique in the proof. The core idea is to construct an FNN with exponential activation functions, incorporating an additional neuron to handle the normalization effect. Detailed proofs are provided in Appendix B. Similar results have been obtained in recent work [12], though via different methodologies.

3 The Non-Universal Approximation Property of $\mathcal{N}_*^\sigma$ and $\mathcal{T}_*^\sigma$

One key aspect of ICL is that the context can act as a control parameter for the model. We now consider the case where the tokens in context is restricted to a finite vocabulary. A natural question arises: can single-layer Transformers with a finite vocabulary, *i.e.*, $\mathcal{T}_*^\sigma$, still achieve the UAP via ICL? We first analyze $\mathcal{N}_*^\sigma$ for simplicity, then using the established connection between FNNs and Transformers to extend the result to $\mathcal{T}_*^\sigma$. The answer is that $\mathcal{N}_*^\sigma$ cannot achieve the UAP because of the restriction of finite parameters.

For element-wise activations, the span of $\mathcal{N}_*^\sigma$, span$\{\mathcal{N}_*^\sigma\}$, forms a finite-dimensional function space. According to results from functional analysis, span$\{\mathcal{N}_*^\sigma\}$ is closed under the function norm (see e.g. Theorem 1.21 of [51] or Corollary C.4 of [52]). This implies that the set of functions approximable by span$\{\mathcal{N}_*^\sigma\}$ is precisely the set of functions within span$\{\mathcal{N}_*^\sigma\}$. Consequently, any function not in span$\{\mathcal{N}_*^\sigma\}$ cannot be arbitrarily approximated, meaning that the UAP cannot be achieved.

For softmax networks, the normalization operation introduces further limitations. Even though $\mathrm{N}_*^{\mathrm{softmax}}$ consists of weighted units drawn from a fixed finite collection of basic units, normalization prevents these networks from being simple linear combinations of one another. While the span of $\mathcal{N}_*^{\mathrm{softmax}}$ might theoretically have infinite dimensionality, its expressive power remains constrained.

To better understand the behavior of functions within $\mathcal{N}_*^{\mathrm{softmax}}$, we present the following proposition as an introduction.

Proposition 5. *The scalar function $h_k(x)=\sum_{i=1}^{k}a_i\mathrm{e}^{b_i x}$, where $a_i,\ b_i,\ x\in\mathbb{R}$ and at least one a_i is nonzero, has at most $k-1$ zero points.*

Proposition 5 establishes the maximum number of zero points for this class of functions. The result can be proved using mathematical induction. The detailed proof is provided in the Appendix C. Then we can summarize the non-universal approximation property of $\mathcal{N}_*^\sigma$ in the following lemma.

Lemma 6. *The function class $\mathcal{N}_*^\sigma$, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function or softmax activation function σ, cannot achieve the UAP. Specifically, for any compact domain $\mathcal{K}\subset\mathbb{R}^{d_x}$, there exists a continuous function $f:\mathcal{K}\to\mathbb{R}^{d_y}$ and $\varepsilon_0>0$ such that*

$$\max_{x\in\mathcal{K}}\left\|f(x)-\mathrm{N}_*^\sigma(\tilde{x})\right\|\geq\varepsilon_0,\quad \forall\,\mathrm{N}_*^\sigma\in\mathcal{N}_*^\sigma. \tag{19}$$

In the proof of Lemma 6, we demonstrated through Proposition 5 that the number of zeros of $\mathrm{N}_*^{\mathrm{softmax}}$ depends solely on a finite set of parameters and constitutes a bounded quantity. Functions can be explicitly constructed whose number of zeros exceeds this bound, thereby preventing their approximation within $\mathcal{N}_*^{\mathrm{softmax}}$.

By leveraging the connection between FNNs and Transformers, we establish Theorem 7 to demonstrate that $\mathcal{T}_*^\sigma$ cannot achieve the UAP.

Theorem 7. *The function class $\mathcal{T}_*^\sigma$, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function or softmax activation function σ and every $\mathrm{T}^\sigma \in \mathcal{T}_*^\sigma$ satisfys Assumption 1, cannot achieve the UAP. Specifically, for any compact domain $\mathcal{K} \subset \mathbb{R}^{d_x-1}$, there exists a continuous function $f : \mathcal{K} \to \mathbb{R}^{d_y}$ and $\varepsilon_0 > 0$ such that*

$$\max_{x\in\mathcal{K}} \left\| f(x) - \mathrm{T}_*^\sigma(\tilde{x}) \right\| \geq \varepsilon_0, \quad \forall\, \mathrm{T}_*^\sigma \in \mathcal{T}_*^\sigma. \tag{20}$$

The result for element-wise activations follows directly from the application of Lemma 3 and Lemma 6. However, the case of the softmax activation requires additional techniques to account for the normalization effect. The proof, which utilizes Proposition 5 once again, is presented in the Appendix C. It is worth noting that Theorem 7 holds even without imposing any constraints on the V, Q and K (e.g., the sparse partition described in equation (14)). Further details can be found in Appendix F.

4 The Universal Approximation Property of $\mathcal{T}_{*,\mathcal{P}}^\sigma$

After establishing that neither $\mathcal{N}_*^\sigma$ nor $\mathcal{T}_*^\sigma$ can achieve the UAP, we aim to leverage a key feature of Transformers: their ability to incorporate APEs during token input. This motivates us to investigate whether $\mathcal{T}_{*,\mathcal{P}}^\sigma$ can realize the UAP.

The answer is affirmative. To support our constructive proof, we invoke the Kronecker Approximation Theorem as a key auxiliary tool. This result ensures the density of certain structured sets in $\mathbb{R}^n$ under mild arithmetic conditions. The formal statement and discussion of this theorem are provided in Appendix D.

Theorem 8. *Let $\mathcal{T}_{*,\mathcal{P}}^\sigma$ be the class of functions $\mathrm{T}_{*,\mathcal{P}}^\sigma$ satisfying Assumption 1, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function σ, the subscript refers the finite vocabulary $\mathcal{V} = \mathcal{V}_x \times \mathcal{V}_y$, $\mathcal{P} = \mathcal{P}_x \times \mathcal{P}_y$ represents the positional encoding map, and denote a set S as:*

$$S := \mathcal{V}_x + \mathcal{P}_x = \left\{ x_i + \mathcal{P}_x^{(j)} \;\middle|\; x_i \in \mathcal{V}_x,\ i,\ j \in \mathbb{N}^+ \right\}. \tag{21}$$

If S is dense in $\mathbb{R}^{d_x}$, $\{1,\ -1,\ \sqrt{2},\ 0\}^{d_y} \subset \mathcal{V}_y$ and $\mathcal{P}_y = 0$, then $\mathcal{T}_{*,\mathcal{P}}^\sigma$ can achieve the UAP. More specifically, given a network $\mathrm{T}_{*,\mathcal{P}}^\sigma$, then for any continuous function $f : \mathbb{R}^{d_x-1} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$ and $\varepsilon > 0$, there always exist $X \in \mathbb{R}^{d_x\times n}$ and $Y \in \mathbb{R}^{d_y\times n}$ from the vocabulary $\mathcal{V}$, i.e. *$x^{(i)} \in \mathcal{V}_x, y^{(i)} \in \mathcal{V}_y$, with some length $n \in \mathbb{N}^+$ such that*

$$\left\| \mathrm{T}_{*,\mathcal{P}}^\sigma\left(\tilde{x}; X, Y\right) - f(x) \right\| < \varepsilon, \quad \forall x \in \mathcal{K}. \tag{22}$$

We provide a constructive proof in Appendix C, and here we only demonstrate the proof idea by considering the specific case of $d_y = 1$ and assuming the matrices U in the Transformer $\mathrm{T}_{*,\mathcal{P}}^\sigma$ is an identity matrice. In this case, the Transformer can be simplified to an FNN N_*^σ, that is

$$\mathrm{T}_{*,\mathcal{P}}^\sigma(x; X, Y) = Y_{\mathcal{P}}\sigma\big(X_{\mathcal{P}}^\top B^\top C\tilde{x}\big) = \sum_{j=1}^{n} y^{(j)}\sigma\Big(\big(x^{(j)} + \mathcal{P}_x^{(j)}\big)B^\top C \cdot \tilde{x}\Big), \tag{23}$$

which is similar to the calculation in equation (17). The UAP of FNNs shown in Lemma 2 implies that the target function f can be approximated by an FNN with k hidden neurons,

$$\mathrm{N}^\sigma(x) = A\sigma(W\tilde{x} + b) = \sum_{i=1}^{k} a_i\sigma(w_i \cdot x + b_i) = \sum_{i=1}^{k} a_i\sigma(\tilde{w}_i \cdot \tilde{x}). \tag{24}$$

Since we are considering a continuous activation function σ, we can conclude that slightly perturbing the parameters A and W will lead to new FNN that can still approximate f. This motivates us to construct a proof using the property that each $\tilde{w}_i \in \mathbb{R}^{d_x}$ can be approximated by vectors $x_{\mathcal{P}} B^\top C, x_{\mathcal{P}} \in S = \mathcal{V}_x + \mathcal{P}_x$, and each $a_i \in \mathbb{R}$ can be approximated by $q_i\sqrt{2} \pm l_i$, with positive integers q_i and l_i.

For ease of exposition, we will first show how to construct X, Y so as to approximate the first term in the summation in equation (24), namely $a_1\sigma\big(\tilde{w}_1 \cdot \tilde{x}\big)$. By lemma 6, we may choose positive integers q and l such that $q\sqrt{2} \pm l$ is sufficiently close to a_1. Consider the first token in the context. Since the positional encoding is fixed, *i.e.* $\mathcal{V}_x + \mathcal{P}^{(1)}$ is a finite set, one of two cases must occur:

1. if there exists a token $x^{(1)} \in \mathcal{V}_x$ for which $x^{(1)} + \mathcal{P}^{(1)}$ is sufficiently close to $\tilde{w}_1$, then we declare this position "valid";

2. otherwise, we declare the position "invalid", and choose any $x^{(1)} \in \mathcal{V}_x$, and set $y^{(1)} = 0$ so as to nullify its contribution in the sum.

We then proceed inductively: having handled the first token, we construct the second token in exactly the same manner, then the third, and so on, until we have identified $q + l$ valid positions. Because S is dense in R^{d_x} and $q,\ l$ are finite, this selection process necessarily terminates after finitely many steps. Finally, we assign $y^{(i)} = \sqrt{2}$ for q of the valid positions and $y^{(i)} = \pm 1$ for other l valid positions. Up to now, we have built a partial context that enables the output of $\mathrm{T}^{\sigma}_{*,\mathcal{P}}$ to approximate $a_1\sigma\big(\tilde{w}_1 \cdot \tilde{x}\big)$ with arbitrarily small error. Once we have approximated $a_1\sigma\big(\tilde{w}_1 \cdot \tilde{x}\big)$, we can in finitely many further steps similarly approximate $a_2\sigma\big(\tilde{w}_2 \cdot \tilde{x}\big), \cdots, a_k\sigma\big(\tilde{w}_k \cdot \tilde{x}\big)$, thereby completing the construction of the full context X and Y. In the proof idea above, we take the density of the set S in $\mathbb{R}^{d_x}$ as a fundamental assumption. $\mathcal{V}_x$ contains only finitely many elements, rendering it bounded. For S to be dense in the entire space, $\mathcal{P}_x$ must be unbounded.

Next, we relax this requirement, eliminating the need for $\mathcal{P}_x$ to be bounded, making the conditions more aligned with practical scenarios. Particularly, we consider the specific activation function in the following theorem, where the notations not explicitly mentioned remain consistent with those in Theorem 8. We present the an informal version, and the formal version is provided in Appendix E.

Theorem 9 (Informal Version)**.** *If the set S is dense in $[-1,1]^{d_x}$, then $\mathcal{T}^{\mathrm{ReLU}}_{*,\mathcal{P}}$ is capable of achieving the UAP. Additionally, if S is only dense in a neighborhood $B(w^*, \delta)$ of a point $w^* \in \mathbb{R}^{d_x}$ with radius $\delta > 0$, then the class of transformers with exponential activation,* i.e. $\mathcal{T}^{\exp}_{*,\mathcal{P}}$*, is capable of achieving the UAP.*

The density condition on S is significantly refined here, which we will discuss in the later remark. This improvement is possible because the proof of Theorem 8 relies directly on the UAP of FNNs, where the weights take values from the entire parameter space. However, for FNNs with specific activations, we can restrict the weights to a small set without losing the UAP.

For ReLU networks, we can use the positive homogeneity property, *i.e.* $A\mathrm{ReLU}(W\tilde{x}) = \lambda^{-1}A\mathrm{ReLU}\big(\lambda W\tilde{x}\big)$ for any $\lambda > 0$, to restrict the weight matrix W. In fact, the restriction that all elements of W take values in the interval $[-1,1]$ does not affect the UAP of ReLU FNNs because the scale of W can be recovered by adjusting the scale of A via choosing a proper λ.

For exponential networks, the condition on S is much weaker than in the ReLU case. This relaxation is nontrivial, and the proof stems from a property of the derivatives of exponential functions. Consider the exponential function $\exp(w \cdot x)$ as a function of $w \in B(w^*, \delta)$, and denote it as $h(w)$,

$$h(w) = \exp(w \cdot x) = \mathrm{e}^{w_1 x_1 + \cdots + w_d x_d}, \quad w,\ x \in \mathbb{R}^d,\ d = d_x, \tag{25}$$

where w_i and $x_i \in \mathbb{R}$ are the components of w and x, respectively. Calculating the partial derivatives of $h(w)$, we observe the following relations:

$$\frac{\partial^\alpha h}{\partial w^\alpha} := \frac{\partial^{|\alpha|} h}{\partial w_1^{\alpha_1} \cdots \partial w_d^{\alpha_d}} = x_1^{\alpha_1} \cdots x_d^{\alpha_d} h(w), \tag{26}$$

where $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ is the index vector representing the order of partial derivatives, and $|\alpha| := \alpha_1 + \cdots + \alpha_d$. This relationship allows us to link exponential FNNs to polynomials since any polynomial $P(x)$ can be represented in the following form:

$$P(x) = \exp\big(-w^* \cdot x\big) \left(\sum_{\alpha \in \Lambda} a_\alpha \frac{\partial^{|\alpha|} h}{\partial w^\alpha} \right) \Bigg|_{w = w^*}, \tag{27}$$

where a_α are the coefficients of the polynomials, Λ is a finite set of indices, and the partial derivatives can be approximated by finite differences, which are FNNs. For example, the first-order partial derivative $\frac{\partial h}{\partial w_1}\big|_{w=w^*} = x_1 h(w^*)$ can be approximated by the following difference with a small nonzero number $\lambda \in (0, \delta)$,

$$\frac{h(w^* + \lambda e_1) - h(w^*)}{\lambda} = \lambda^{-1} \exp((w^* + \lambda e_1) \cdot x) - \lambda^{-1} \exp(w^* \cdot x). \tag{28}$$

This is an exponential FNN with two neurons. Finally, employing the well-known Stone-Weierstrass theorem, which states that any continuous function f on compact domains can be approximated by polynomials, and combining the above relations between FNNs and polynomials, we can establish the UAP of exponential FNNs with weight constraints.

Remark 10. *When discussing density, one of the most immediate examples that comes to mind is the density of rational numbers in $\mathbb{R}$. How can we effectively enumerate rational numbers? The work by [53] introduces an elegant method for enumerating positive rational numbers, synthesizing ideas from [54] and [55]. It demonstrates the computational feasibility of enumeration through an effective algorithm. Thus, we assume that positional encodings can be implemented using computer algorithms, such as iterative functions.*

5 Conclusion

In this paper, we establish a connection between FNNs and Transformers through ICL. By leveraging the UAP of FNNs, we demonstrate that the UAP of ICL holds when the context is selected from the entire vector space. When the context is drawn from a finite set, we explore the approximation power of VICL, showing that the UAP is achievable only when appropriate positional encodings are incorporated, underscoring the importance of positional encodings.

In our work, we consider Transformers with input sequences of arbitrary length, implying that the positional encoding $\mathcal{P}_x$ consists of a countably infinite set of elements. In Theorem 8, we assume a strong density condition, which is later relaxed in Theorem 9. However, in practical applications, input sequences are finite, typically truncated for computational feasibility. This shift allows our conclusions to be interpreted through an approximation lens, where the objective is to approximate functions within a specified error margin, rather than achieving infinitesimal precision. Additionally, to achieve the UAP, it is insightful to compare the function approximation capabilities of our approach (outlined in Lemma 4) with the direct use of FNNs, particularly when the Transformer parameters are trainable.

It is important to note that this paper is limited to single-layer Transformers with APEs, and the main results (Theorem 8 and Theorem 9) focus on element-wise activations. Future research should extend these findings to multi-layer Transformers, general positional encodings (such as RPEs and RoPE), and softmax activations. For softmax Transformers, our analysis in Sections 2 and 3 highlighted their connection to Transformers with exponential activations. However, extending this connection to the scenario in Section 4 proves challenging and requires more sophisticated techniques.

Although this paper primarily addresses theoretical issues, we believe our results can offer valuable insights for practitioners. Specifically, in Remark 10, we observe that certain algorithms use function composition to enumerate numbers dense in $\mathbb{R}$. This idea could inspire the design of positional encodings via compositions of fixed functions, similar to RNN approaches. RNNs capture the sequential nature of information by integrating the importance of word order in sentence meaning. However, to the best of our knowledge, existing research on RNNs has not explored the denseness properties of the sets formed by their hidden state sequences. We hope this unexplored property will inspire experimental research in future studies. Furthermore, our construction for Theorem 8 relies on the sparse partition assumption in equation (14). The practical validity of this assumption remains uncertain, and we leave this question open for future exploration.

In fact, [56, 57] on continuous CoTs and continuous states have certain connections to our work – specifically, leveraging positional encoding to enable Transformers to achieve the UAP for functions whose domain is a finite set while the range covers the entire Euclidean space. Moreover, Xiao et al. [58] proposing an approach for automatically adjusting prompts for function fitting is also related to our theoretical findings. Therefore, with further research, our theory holds practical significance.

References

[1] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, L. Li, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2024.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020.

[3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, 2022.

[4] Z. Chu, J. Chen, Q. Chen, W. Yu, T. He, H. Wang, W. Peng, M. Liu, B. Qin, and T. Liu, "Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future," in *Annual Meeting of the Association for Computational Linguistics*, 2024.

[5] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2024.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, pp. 303–314, 1989.

[7] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[8] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, pp. 251–257, 1991.

[9] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.

[10] C. Yun, S. Bhojanapalli, A. S. Rawat, S. Reddi, and S. Kumar, "Are transformers universal approximators of sequence-to-sequence functions?" in *International Conference on Learning Representations*, 2020.

[11] S. Luo, S. Li, S. Zheng, T.-Y. Liu, L. Wang, and D. He, "Your transformer may not be as powerful as you expect," in *Advances in Neural Information Processing Systems*, 2022.

[12] A. Petrov, P. Torr, and A. Bibi, "Prompting a pretrained transformer can be a universal approximator," in *International Conference on Machine Learning*, 2024.

[13] K. Ahn, X. Cheng, H. Daneshmand, and S. Sra, "Transformers learn to implement preconditioned gradient descent for in-context learning," in *Advances in Neural Information Processing Systems*, 2024.

[14] X. Cheng, Y. Chen, and S. Sra, "Transformers implement functional gradient descent to learn non-linear functions in context," in *International Conference on Machine Learning*, 2024.

[15] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems*, 2017.

[16] S. Park, C. Yun, J. Lee, and J. Shin, "Minimum width for universal approximation," in *International Conference on Learning Representations*, 2021.

[17] Y. Cai, "Achieve the minimum width of neural networks for universal approximation," in *International Conference on Learning Representations*, 2023.

[18] L. Li, Y. Duan, G. Ji, and Y. Cai, "Minimum width of leaky-relu neural networks for uniform universal approximation," in *arXiv:2305.18460v3*, 2024.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. ukasz Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[21] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, 2019.

[22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[23] L. Zhenzhong, C. Mingda, G. Sebastian, G. Kevin, S. Piyush, and S. Radu, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2021.

[24] X. Liu, H.-F. Yu, I. Dhillon, and C.-J. Hsieh, "Learning to encode position for transformer with continuous dynamical model," in *International Conference on Machine Learning*, 2020.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[27] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, pp. 1–113, 2023.

[28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[29] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2024.

[30] G. Xun, X. Jia, V. Gopalakrishnan, and A. Zhang, "A survey on context learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, pp. 38–56, 2017.

[31] S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, “Want to reduce labeling cost? gpt-3 can help,” in *Empirical Methods in Natural Language Processing*, 2021.

[32] H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, and S. Groppe, “Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text,” *arXiv preprint arXiv:2305.08804*, 2023.

[33] B. Ding, C. Qin, L. Liu, Y. K. Chia, B. Li, S. Joty, and L. Bing, “Is gpt-3 a good data annotator?” in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[34] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, “In-context retrieval-augmented language models,” in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[35] N. De Cao, W. Aziz, and I. Titov, “Editing factual knowledge in language models,” in *Empirical Methods in Natural Language Processing*, 2021.

[36] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” in *International Conference on Learning Representations*, 2021.

[37] B. Wang, L. Shang, C. Lioma, X. Jiang, H. Yang, Q. Liu, and J. G. Simonsen, “On position embeddings in bert,” in *International Conference on Learning Representations*, 2021.

[38] G. Ke, D. He, and T.-Y. Liu, “Rethinking positional encoding in language pre-training,” in *International Conference on Learning Representations*, 2021.

[39] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Annual Meeting of the Association for Computational Linguistics*, 2018.

[40] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[41] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “Roformer: Enhanced transformer with rotary position embedding,” *Neurocomputing*, vol. 568, p. 127063, 2024.

[42] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, “An explanation of in-context learning as implicit bayesian inference,” in *International Conference on Learning Representations*, 2022.

[43] X. Wang, W. Zhu, M. Saxon, M. Steyvers, and W. Y. Wang, “Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning,” in *Advances in Neural Information Processing Systems*, 2024.

[44] D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei, “Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers,” in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[45] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” in *International Conference on Learning Representations*, 2022.

[46] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” in *Advances in Neural Information Processing Systems*, 2022.

[47] S. Alberti, N. Dern, L. Thesing, and G. Kutyniok, “Sumformer: Universal approximation for efficient transformers,” *Annual Workshop on Topology, Algebra, and Geometry in Machine Learning*, pp. 72–86, 2023.

[48] V. Likhosherstov, K. Choromanski, and A. Weller, “On the expressive power of self-attention matrices,” *arXiv preprint arXiv:2106.03764*, 2021.

[49] P. Deora, R. Ghaderi, H. Taheri, and C. Thrampoulidis, “On the optimization and generalization of multi-head attention,” *Transactions on Machine Learning Research*, 2024.

[50] A. Petrov, P. Torr, and A. Bibi, “When do prompting and prefix-tuning work? a theory of capabilities and limitations,” in *International Conference on Learning Representations*, 2024.

[51] W. Rudin, *Functional Analysis*. McGraw-Hill Science, 1991.

[52] P. Cannarsa and T. D’Aprile, *Introduction to Measure Theory and Functional Analysis*. Springer Cham, 2015.

[53] N. J. Calkin and H. S. Wilf, “Recounting the rationals,” *The American Mathematical Monthly*, vol. 107, pp. 360–363, 2000.

[54] M. Stern, “Ueber eine zahlentheoretische funktion,” *Journal für die reine und angewandte Mathematik*, vol. 1858, pp. 193–220, 1858.

[55] B. C. Berndt, H. G. Diamond, H. Halberstam, and A. Hildebrand, *Analytic Number Theory: Proceedings of a Conference in Honor of Paul T. Bateman*. Birkhäuser, 1990.

[56] J. Tack, J. Lanchantin, J. Yu, A. Cohen, I. Kulikov, J. Lan, S. Hao, Y. Tian, J. Weston, and X. Li, “Llm pretraining with continuous concepts,” *arXiv preprint arXiv:2502.08524*, 2025.

[57] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, Tian, and Yuandong, “Training large language models to reason in a continuous latent space,” *arXiv preprint arXiv:2412.06769*, 2024.

[58] T. Z. Xiao, R. Bamler, B. Schölkopf, and W. Liu, “Verbalized machine learning: Revisiting machine learning with language models,” *Transactions on Machine Learning Research*, 2025.

[59] T. M. Apostol, *Modular Functions and Dirichlet Series in Number Theory (Graduate Texts in Mathematics, 41)*. Springer, 1989.

[60] G. Boole, *A Treatise on the Calculus of Finite Differences*. Cambridge University Press, 2009.

A Table of Notations

We present all our notations in Table 1 for easy reference.

Table 1: Table of Notations

Notations	Explanations
$d_x,\ d_y$	Dimensions of input and output.
$\mathcal{P}$	Positional encoding.
X, Y	Context without positional encoding.
$X_{\mathcal{P}},\ Y_{\mathcal{P}}$	Context with positional encoding $\mathcal{P}$.
Z	Input without positional encoding.
$Z_{\mathcal{P}}$	Input with positional encoding $\mathcal{P}$.
$\mathcal{V}$	Vocabulary.
$\mathcal{V}_x,\ \mathcal{V}_y$	Vocabulary of $x^{(i)}$ and $y^{(i)}$.
σ	Activation function.
$\#$	The cardinality of a set.
$\mathrm{N}^\sigma,\ \mathcal{N}^\sigma$	One-hidden-layer FNN and its collection.
$\mathrm{T}^\sigma,\ \mathcal{T}^\sigma$	Single-layer Transformer and its collection.
$\mathrm{N}^\sigma_*,\ \mathcal{N}^\sigma_*$	One-hidden-layer FNN with a finite set of weights and its collection.
$\mathrm{T}^\sigma_*,\ \mathcal{T}^\sigma_*$	Single-layer Transformer with vocabulary restrictions and its collection.
$\mathrm{T}^\sigma_{*,\mathcal{P}},\ \mathcal{T}^\sigma_{*,\mathcal{P}}$	Single-layer Transformer with positional encoding, vocabulary restrictions, and its collection.
$\\|\cdot\\|$	The uniform norm of vectors, *i.e.*, a shorthand for $\\|\cdot\\|_\infty$.
$\tilde{x}$	Append a one to the end of x, *i.e.*, $\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$.

B Proofs of Section 2

We provide detailed proofs of lemmas in Section 2. We will first directly proof Lemma 3 using Lemma 2. Next, by a similar method together with an additional technical refinement, we will establish Lemma 12. Finally, leveraging Lemma 12, we will prove Lemma 4.

B.1 Proof of Lemma 3

Lemma 3. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-polynomial, locally bounded, piecewise continuous activation function, and T^σ be a single-layer Transformer satisfying Assumption 1. For any one-hidden-layer network $\mathrm{N}^\sigma : \mathbb{R}^{d_x-1} \to \mathbb{R}^{d_y} \in \mathcal{N}^\sigma$ with n hidden neurons, there exist matrices $X \in \mathbb{R}^{d_x \times n}$ and $Y \in \mathbb{R}^{d_y \times n}$ such that*

$$\mathrm{T}^\sigma\left(\tilde{x}; X, Y\right) = \mathrm{N}^\sigma(x), \quad \forall x \in \mathbb{R}^{d_x-1}. \tag{29}$$

Proof. We can directly compute the output of T^σ is

$$\begin{aligned} \mathrm{T}^\sigma(\tilde{x}, X, Y) &= \left(Z + \mathrm{Attn}^\sigma_{Q,K,V}(\tilde{x}, X, Y)\right)_{d_x+1:d_x+d_y, n+1} \\ &= \left(Z + VZM\sigma(Z^\top Q^\top K Z)\right)_{d_x+1:d_x+d_y, n+1} \\ &= \left(Z + \begin{bmatrix} DX + Ey & 0 \\ UY & 0 \end{bmatrix} \begin{bmatrix} \sigma(X^\top B^\top C X) & \sigma(X^\top B^\top C\tilde{x}) \\ \sigma(\tilde{x}^\top B^\top C X) & \sigma(\tilde{x}^\top B^\top C \tilde{x}) \end{bmatrix}\right)_{d_x+1:d_x+d_y, n+1} \\ &= UY\sigma(X^\top B^\top C\tilde{x}). \end{aligned} \tag{30}$$

One can easily observe that the output closely resembles that of a single-layer FNN. Suppose $\mathrm{N}^\sigma(x) = A\sigma(Wx + b) : \mathbb{R}^{d_x-1} \to \mathbb{R}^{d_y}$ is an arbitrary single-layer FNN with k hidden neurons, where and $W \in \mathbb{R}^{k \times (d_x-1)},\ A \in \mathbb{R}^{d_y \times k},\ b \in \mathbb{R}^k$. We construct the context by setting its length to k, *i.e.* $X \in \mathbb{R}^{d_x \times k},\ Y \in \mathbb{R}^{d_y \times k}$. Then, through straightforward calculation, we find that choosing

$$X = (C^\top B)^{-1} \begin{bmatrix} W & b \end{bmatrix}^\top, \quad Y = U^{-1} A, \tag{31}$$

is sufficient to ensure that $\mathrm{T}^{\sigma}(\tilde{x}; X, Y) = \mathrm{N}^{\sigma}(x)$. □

Remark 11. *It is worth noting that in the above proof, the matrix F was set to zero in accordance with Assumption 1. However, we emphasize that this is not a strict requirement. In fact, one can accommodate arbitrary F by choosing $Y = U^{-1}(A - FX)$. The choice $F = 0$ is made purely for computational convenience under our current assumptions.*

B.2 Proof of the UAP of Softmax FNNs

Before proving Lemma 4, we demonstrate the UAP of softmax FNNs as a supporting lemma.

Lemma 12 (UAP of Softmax FNNs)**.** *For any continuous function $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$, and for any $\varepsilon > 0$, there exist a network $\mathrm{N}^{\mathrm{softmax}}(x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ satisfying*

$$\|\mathrm{N}^{\mathrm{softmax}}(x) - f(x)\| < \varepsilon, \quad \forall x \in \mathcal{K}. \tag{32}$$

Proof. We first build a bridge connecting softmax FNNs and target function f according to Theorem 2. We can construct a network

$$\mathrm{N}^{\exp}(x) = A\exp(Wx + b) = \sum_{i=1}^{k} a_i \mathrm{e}^{w_i \cdot x + b_i}, \tag{33}$$

with k hidden neurons satisfying

$$\max_{x \in \mathcal{K}} \|\mathrm{N}^{\exp}(x) - f(x)\| < \frac{\varepsilon}{2}, \tag{34}$$

where $a_i \in \mathbb{R}^{d_y},\ w_i \in \mathbb{R}^{d_x},\ b_i \in \mathbb{R}$. Then we only need to construct a softmax FNN $\mathrm{N}^{\mathrm{softmax}}(x)$ which can approximate such $\mathrm{N}^{\exp}(x)$, and this can be succinctly described as seeking a method to eliminate the effects of normalization.

Consider a softmax FNN

$$\mathrm{N}^{\mathrm{softmax}}(x) = A' \operatorname{softmax}\left(W'x + b'\right) = \frac{\sum_{i=1}^{k+1} a_i' \mathrm{e}^{w_i' \cdot x + b_i'}}{\sum_{j=1}^{k+1} \mathrm{e}^{w_j' \cdot x + b_j'}}, \tag{35}$$

with $k+1$ hidden neurons, where $w_{k+1}' = b_{k+1}' = 0$, $b_i' = b_i'(\varepsilon)$ is sufficiently small to satisfy

$$\mathrm{e}^{w_i' \cdot x + b_i'} < \frac{\varepsilon}{2k\big(1 + \max_{x \in \mathcal{K}} \|\mathrm{N}^{\exp}(x)\|\big)}, \quad \forall x \in \mathcal{K},\ i = 1, 2, \cdots, k, \tag{36}$$

and $w_i' = w_i$ for $i = 1, 2, \cdots, k$. This arrangement ensures that, in the denominators of each term in Equation (35), the first k entries are arbitrarily small, while the $(k+1)$-th entry is exactly one. We then simply adjust A' so that the numerators coincide with those in Equation (33), and this can be done by setting $a_i' = \begin{cases} a_i \mathrm{e}^{b_i - b_i'}, & i = 1, 2, \cdots, k \\ 0, & i = k+1 \end{cases}$. With the formal construction now complete, we present a more precise estimate of the approximation error as follows.

$$
\begin{aligned}
\|\mathrm{N}^{\exp}(x) - \mathrm{N}^{\mathrm{softmax}}(x)\| &= \max_{x\in\mathcal{K}} \left\| \sum_{i=1}^{k} a_i \mathrm{e}^{w_i\cdot x + b_i} - \frac{\sum_{i=1}^{k+1} a_i' \mathrm{e}^{w_i'\cdot x + b_i'}}{\sum_{j=1}^{k+1} \mathrm{e}^{w_j'\cdot x + b_j'}} \right\| \\
&= \max_{x\in\mathcal{K}} \left\| \sum_{i=1}^{k} a_i \mathrm{e}^{w_i\cdot x + b_i} - \frac{\sum_{i=1}^{k} a_i \mathrm{e}^{w_i\cdot x + b_i}}{\sum_{j=1}^{k} \mathrm{e}^{w_j'\cdot x + b_j'} + 1} \right\| \\
&= \max_{x\in\mathcal{K}} \|\mathrm{N}^{\exp}(x)\| \cdot \max_{x\in\mathcal{K}} \left\| 1 - \frac{1}{\sum_{j=1}^{k} \mathrm{e}^{w_j'\cdot x + b_j'} + 1} \right\| \\
&\le \max_{x\in\mathcal{K}} \|\mathrm{N}^{\exp}(x)\| \cdot \max_{x\in\mathcal{K}} \left\| \sum_{j=1}^{k} \mathrm{e}^{w_j'\cdot x + b_j'} \right\| \\
&\le \frac{\varepsilon}{2}.
\end{aligned}
\tag{37}
$$

This leads to the conclusion that $\|\mathrm{N}^{\mathrm{softmax}}(x) - f(x)\| < \varepsilon$ for all $x \in \mathcal{K}$, which finishes the proof. □

B.3 Proof of Lemma 4

Lemma 4. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-polynomial, locally bounded, piecewise continuous activation function or softmax function, and T^{σ} be a single-layer Transformer satisfying Assumption 1, and $\mathcal{K}$ be a compact domain in $\mathbb{R}^{d_x-1}$. Then for any continuous function $f : \mathcal{K} \to \mathbb{R}^{d_y}$ and any $\varepsilon > 0$, there exist matrices $X \in \mathbb{R}^{d_x\times n}$ and $Y \in \mathbb{R}^{d_y\times n}$ such that*

$$
\left\|\mathrm{T}^{\sigma}\left(\tilde{x}; X, Y\right) - f(x)\right\| < \varepsilon, \quad \forall x \in \mathcal{K}. \tag{38}
$$

Proof. For element-wise activation cases, with the help of Theorem 2 and Lemma 3, the conclusion follows trivially.

Then we solve the softmax case. Similarly, for any $\varepsilon > 0$, we can construct a softmax FNN $\mathrm{N}^{\mathrm{softmax}}(x)$ with k hidden neurons, using Lemma 12 such that

$$
\max_{x\in\mathcal{K}} \|\mathrm{N}^{\mathrm{softmax}}(x) - f(x)\| < \frac{\varepsilon}{2}. \tag{39}
$$

Then what we need to do is to approximate this softmax FNN with a softmax transformer. We can directly compute the following

$$
\begin{aligned}
&\mathrm{T}^{\mathrm{softmax}}(\tilde{x}, X, Y) \\
&= \left(Z + \begin{bmatrix} DX + EY & 0 \\ UY & 0 \end{bmatrix} \operatorname{softmax}\left(\begin{bmatrix} X^\top B^\top C X & X^\top B^\top C \tilde{x} \\ \tilde{x}^\top B^\top C X & \tilde{x}^\top B^\top C \tilde{x} \end{bmatrix} \right) \right)_{d_x+1:d_x+d_y,\, n+1} \\
&= UY \left(\operatorname{softmax}\left(\begin{bmatrix} X^\top B^\top C \tilde{x} \\ \tilde{x}^\top B^\top C \tilde{x} \end{bmatrix} \right) \right)_{1:n}.
\end{aligned}
\tag{40}
$$

Through comparing the output with the exponential FNN, we can find out that there is one more bounded positive term $t(x) := \exp\left(\tilde{x}^\top B^\top C \tilde{x}\right)$ when processing normalization.

Chose the length of context $n = k + 1$ and $X,\ Y$ such that

$$
X^\top B^\top C = \begin{bmatrix} W & b + s\mathbf{1} \\ 0 & s \end{bmatrix},\ UY = \begin{bmatrix} A & 0 \end{bmatrix} \tag{41}
$$

where $\mathbf{1}$ is all-ones vector and s is big enough, making

$$\mathrm{e}^{\tilde{x}^\top B^\top C\tilde{x}-s} < \frac{\varepsilon}{2\big(1+\max_{x\in\mathcal{K}}\|\mathrm{N}^{\mathrm{softmax}}(x)\|\big)}, \quad \forall x \in \mathcal{K}. \tag{42}$$

Then $X^\top B^\top C\tilde{x} = \begin{bmatrix} W & b+s\mathbf{1} \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} Wx+b+s\mathbf{1} \\ s \end{bmatrix}$, and we can compute a detailed form of $\mathrm{T}^{\mathrm{softmax}}(\tilde{x}; X, Y)$ as:

$$\begin{aligned} \mathrm{T}^{\mathrm{softmax}}(\tilde{x}; X, Y) &= \frac{\sum_{i=1}^{k} a_i \exp(w_i \cdot x + b_i + s)}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j + s) + \exp(s) + \exp(\tilde{x}^\top B^\top C\tilde{x})} \\ &= \frac{\sum_{i=1}^{k} a_i \exp(w_i \cdot x + b_i)}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1 + \exp(\tilde{x}^\top B^\top C\tilde{x} - s)}. \end{aligned} \tag{43}$$

We focus on estimating the upper bound of the distance between $\mathrm{N}^{\mathrm{softmax}}$ and $\mathrm{T}^{\mathrm{softmax}}$, that is

$$\begin{aligned} &\max_{x\in\mathcal{K}} \|\mathrm{N}^{\mathrm{softmax}}(x) - \mathrm{T}^{\mathrm{softmax}}(\tilde{x}; X, T)\| \\ &= \max_{x\in\mathcal{K}} \left\| \frac{\sum_{i=1}^{k} a_i \exp(w_i \cdot x + b_i)}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1} - \frac{\sum_{i=1}^{k} a_i \exp(w_i \cdot x + b_i)}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1 + \exp(\tilde{x}^\top B^\top C\tilde{x} - s)} \right\| \\ &= \max_{x\in\mathcal{K}} \|\mathrm{N}^{\mathrm{softmax}}(x)\| \cdot \max_{x\in\mathcal{K}} \left\| 1 - \frac{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1 + \exp(\tilde{x}^\top B^\top C\tilde{x} - s)} \right\| \\ &= \max_{x\in\mathcal{K}} \|\mathrm{N}^{\mathrm{softmax}}(x)\| \cdot \max_{x\in\mathcal{K}} \left\| \frac{\exp(\tilde{x}^\top B^\top C\tilde{x} - s)}{\sum_{j=1}^{k} \exp(w_j \cdot x + b_j) + 1 + \exp(\tilde{x}^\top B^\top C\tilde{x} - s)} \right\| \\ &\le \max_{x\in\mathcal{K}} \|\mathrm{N}^{\mathrm{softmax}}(x)\| \cdot \max_{x\in\mathcal{K}} \big\| \exp(\tilde{x}^\top B^\top C\tilde{x} - s) \big\| \\ &\le \frac{\varepsilon}{2}. \end{aligned} \tag{44}$$

As a consequence, we have $\big\|\mathrm{T}^{\sigma}\big(\tilde{x}; X, Y\big) - f(x)\big\| < \varepsilon$ for all $x \in \mathcal{K}$, which finishes the proof. □

C Proofs of Section 3

In this appendix, we provide detailed proofs of Proposition 5, Lemma 6 and Theorem 7 presented in Section 3. We will first using induction to prove Proposition 5, then employ this proposition together with a proof by contradiction to establish Lemma 6 and Theorem 7.

C.1 Proof of Proposition 5

Proposition 5. *The scalar function* $h_k(x) = \sum_{i=1}^{k} a_i \mathrm{e}^{b_i x}$*, where* $a_i,\ b_i,\ x \in \mathbb{R}$ *and at least one* a_i *is nonzero, has at most* $k-1$ *zero points.*

Proof. We prove this statement by induction. When $k = 1$ and 2, the statement is easy to prove. We suppose $h_N(x)$ has at most $N-1$ zero points, now consider the case $k = N+1$. Let

$h_{N+1}(x) = \sum_{i=1}^{N+1} a_i \mathrm{e}^{b_i x}$. Without loss of generality, assume that $a_{N+1} \neq 0$. Thus, we can rewrite $h_{N+1}(x)$ as

$$h_{N+1}(x) = a_{N+1}\mathrm{e}^{b_{N+1}x}\big(1 + \sum_{i=1}^{N} \frac{a_i}{a_N+1}\mathrm{e}^{(b_i - b_{N+1})x}\big) := a_{N+1}\mathrm{e}^{b_{N+1}x} g(x).$$

Then we process by contradiction. Suppose $h_{N+1}(x)$ has more than N zero points, which implies $g(x)$ has more than N zero points. Then, according to Rolle's Theorem, $g'(x)$ must have more than $N-1$ zero points, which contradicts our assumption. Thus, h_{N+1} have at most N zero points, and the proof is complete. □

C.2 Proof of Lemma 6

Lemma 6. *The function class $\mathcal{N}_*^\sigma$, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function or softmax activation function σ, cannot achieve the UAP. Specifically, for any compact domain $\mathcal{K} \subset \mathbb{R}^{d_x}$, there exists a continuous function $f : \mathcal{K} \to \mathbb{R}^{d_y}$ and $\varepsilon_0 > 0$ such that*

$$\max_{x\in\mathcal{K}} \big\| f(x) - \mathrm{N}_*^\sigma(\tilde{x}) \big\| \geq \varepsilon_0, \quad \forall\, \mathrm{N}_*^\sigma \in \mathcal{N}_*^\sigma. \tag{45}$$

Proof. For any element-wise activations σ, $\mathrm{span}\{\mathcal{N}^\sigma\}$, forms a finite-dimensional function space. $\mathrm{Span}\{\mathcal{N}^\sigma\}$ is closed under the uniform norm supported by Theorem 2.1 from [51] and Corollary C.4 from [52]. This implies that the set of functions approximable by $\mathrm{span}\{\mathcal{N}^\sigma\}$ is precisely the set of functions within $\mathrm{span}\{\mathcal{N}^\sigma\}$. Consequently, any function not in $\mathrm{span}\{\mathcal{N}^\sigma\}$ cannot arbitrarily approximated, meaning that the UAP cannot be achieved.

Then we prove the softmax case. First, we simplify the problem to facilitate the construction of a function that cannot be approximated. We observe that it suffices to prove the UAP fails when the first input coordinate ranges over $[0, 1]$ and all other coordinates are held fixed. Indeed, for any compact set $K \subset \mathbb{R}^{d_x}$, we can find a closed cube $\prod_{i=1}^{d_x}[l_i, r_i] \subset K$. If we can show that $\mathcal{N}^{\mathrm{softmax}}$ does not achieve the UAP on $[l_1, r_1] \times \prod_{i=2}^{d_x}\{l_i\}$, then, by applying a suitable affine change of variables, it follows that UAP also fails on $[0, 1] \times \prod_{i=2}^{d_x}\{l_i\}$. Consider a continuous target function

$$f : [0, 1] \times \prod_{i=2}^{d_x}\{l_i\} \to \mathbb{R},\ (x_1, x_2, \cdots, x_{d_x}) \mapsto f_1(x_1). \tag{46}$$

The reason why we consider such target function is that every vector-value function $f(x_1, \cdots, x_{d_x})$ can be represent as $f(x_1, \cdots, x_{d_x}) = \big(f_1(x_1, \cdots, x_{d_x}), \cdots, f_{d_y}(x_1, \cdots, x_{d_x})\big)$. If the UAP fails for f, it must fail on at least one of its scalar components. Hence it suffices to consider the one-dimensional (scalar) case. Moreover, since the values of $x_2, \cdots, x_{d_x}$ are fixed, the above reduction to a single-variable scalar function is justified. We only need to demonstrate that there exists at least one such function that cannot be approximated arbitrarily well by any $\mathrm{N}_*^{\mathrm{softmax}} \in \mathcal{N}_*^{\mathrm{softmax}}$.

Then we will use Proposition 5 to finish the rest part of this proof. Before that, we need to rewrite the form of the output of $\mathrm{N}^{\mathrm{softmax}}$, which is

$$\mathrm{N}_*^{\mathrm{softmax}}(x) = \frac{\sum\limits_{i=1}^{k} a_i \mathrm{e}^{w_i\cdot x_i + b_i}}{\sum\limits_{j=1}^{k} \mathrm{e}^{w_j\cdot x_j + b_j}}, \tag{47}$$

where $(a_i, w_i, b_i) \in \mathcal{A} \times \mathcal{W} \times \mathcal{B}$ is a finite set and k is the number of hidden neurons. Consequently, the set $\{\mathcal{W} \times \mathcal{B}\}$ is finite, and we denote it as $N := \#\{\mathcal{W} \times \mathcal{B}\}$. By regrouping identical terms in the numerator, we can rewrite the equation as

$$\mathrm{N}_*^{\mathrm{softmax}}(x) = \frac{\sum\limits_{i=1}^{N} \tilde{a}_i \mathrm{e}^{w_i\cdot x_i + b_i}}{\sum\limits_{j=1}^{d_x} \mathrm{e}^{w_j\cdot x_j + b_j}}. \tag{48}$$

It is important to note that this transformation applies to any $\mathrm{N}_*^{\mathrm{softmax}} \in \mathcal{N}_*^{\mathrm{softmax}}$, ensuring that the number of summation terms in the numerator remains strictly bounded by N.

Finally, we construct a function which cannot be approximated by such softmax networks. Assume a continuous target function

$$g : [0,1] \times \prod_{i=2}^{d_x} \{l_i\} \to \mathbb{R},\ (x_1, x_2, \cdots, x_{d_x}) \mapsto \cos\big((N+1)\pi x_1\big), \tag{49}$$

who has $(N+1)$ zero points in. If $\mathcal{N}_*^{\mathrm{softmax}}$ achieves the UAP, we assume that $\mathrm{N}_*^{\mathrm{softmax}} \in \mathcal{N}_*^{\mathrm{softmax}}$ which satisfies $\|\mathrm{N}_*^{\mathrm{softmax}} - g\| \leq \varepsilon < \frac{1}{10}$. We denote $z_i = \frac{i}{N+1}$ for $i = 0, 1, \cdots, N+1$. It easy to find out that $g(z_i) = 1$ if i is even, and $g(z_i) = -1$ if i is odd, which means $\mathrm{N}_*^{\mathrm{softmax}}(z_i) > 0.9$ for even i is and $\mathrm{N}_*^{\mathrm{softmax}}(z_i) < -0.9$ for odd i. According to Rolle's Theorem, $\mathrm{N}_*^{\mathrm{softmax}}$ has at least $N+1$ zero points, which is contradicts to the Proposition 5. And we finish our proof. $\square$

We will use Figure 1 to provide readers with an intuitive illustration of why a class of functions whose number of zeros is bounded cannot achieve universal approximation.

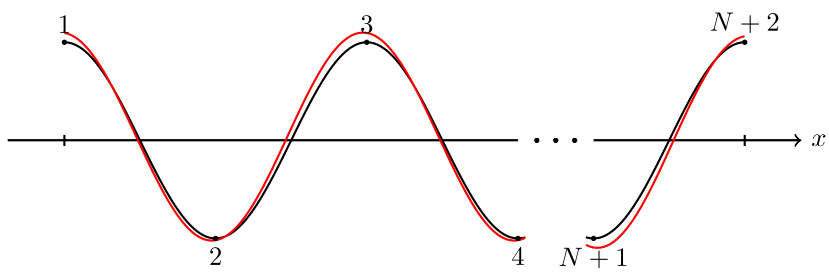

Figure 1: A demonstration of function cannot be approximate. The black curve represents the target function, which has $N+1$ zero points. The red curve represents a sum of exponentials, which has no more then N zero points. If the UAP holds, then the red curve must pass near the $N+2$ marked extrema in the figure. By Rolle's theorem, the function represented by the red curve would then have $N+1$ zeros, which contradicts its intrinsic properties.

C.3 Proof of Theorem 7

Theorem 7. *The function class $\mathcal{T}_*^{\sigma}$, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function or softmax activation function σ and every $\mathrm{T}^{\sigma} \in \mathcal{T}_*^{\sigma}$ satisfys Assumption 1, cannot achieve the UAP. Specifically, for any compact domain $\mathcal{K} \subset \mathbb{R}^{d_x - 1}$, there exists a continuous function $f : \mathcal{K} \to \mathbb{R}^{d_y}$ and $\varepsilon_0 > 0$ such that*

$$\max_{x \in \mathcal{K}} \big\|f(x) - \mathrm{T}_*^{\sigma}(\tilde{x})\big\| \geq \varepsilon_0, \quad \forall\, \mathrm{T}_*^{\sigma} \in \mathcal{T}_*^{\sigma}. \tag{50}$$

Proof. For cases of element-wise activation, since T_*^{σ} has a similar structure to N_*^{σ}, we find that $\mathrm{span}\{\mathrm{T}_*^{\sigma}\}$ is also a finite-dimensional function space. Hence, the same argument from Lemma 6 can be applied here to complete the proof.

Then we prove the softmax case. Recall Equation (40), the output of $\mathrm{T}_*^{\mathrm{softmax}}(\tilde{x}; X, Y)$ can be view as

$$\mathrm{T}_*^{\mathrm{softmax}}(\tilde{x}; X, Y) = \frac{\sum_{i=1}^{n} a_i \mathrm{e}^{w_i \cdot x_i + b_i}}{\sum_{j=1}^{n} \mathrm{e}^{w_j \cdot x_j + b_j} + \mathrm{e}^{\tilde{x}^\top B^\top C \tilde{x}}}, \tag{51}$$

where n represents the length of context and $a_i \in \mathcal{A},\ w_i \in \mathcal{W},\ b_i \in \mathcal{B}$ for some finite sets $\mathcal{A},\ \mathcal{W},\ \mathcal{B}$. This allow us to apply the same approach then proving Lemma 6, which leads to the conclusion that $\mathcal{T}_*^{\sigma}$ cannot achieve the UAP. $\square$

D Kronecker Approximation Theorem

To facilitate our constructive proof, we introduce the Kronecker Approximation Theorem as an auxiliary tool to support the main theorem.

Lemma 13 (Kronecker Approximation Theorem [59])**.** *Given real n-tuples $\alpha^{(i)} = (\alpha_1^{(i)}, \alpha_2^{(i)}, \cdots, \alpha_n^{(i)}) \in \mathbb{R}^n$ for $i = 1, \cdots, m$ and $\beta = (\beta_1, \beta_2, \cdots, \beta_n) \in \mathbb{R}^n$, the following condition holds: for any $\varepsilon > 0$, there exist $q_i, l_i \in \mathbb{Z}$ such that*

$$\left\| \beta_j - \sum_{i=1}^{m} q_i \alpha_j^{(i)} + l_j \right\| < \varepsilon, \quad j = 1, \cdots, n, \tag{52}$$

if and only if for any $r_1, \cdots, r_n \in \mathbb{Z},\ i = 1, \cdots, m$ with

$$\sum_{j=1}^{n} \alpha_j^{(i)} r_j \in \mathbb{Z}, \quad i = 1, \cdots, m, \tag{53}$$

the number $\sum_{j=1}^{n} \beta_j r_j$ is also an integer. In the case of $m = 1$ and $n = 1$, for any $\alpha,\ \beta \in \mathbb{R}$ with α irrational and $\varepsilon > 0$, there exist integers l and q with $q > 0$ such that $|\beta - q\alpha + l| < \varepsilon$.

Lemma 13 indicates that if the condition in equation (53) is satisfied only when all r_i are zeros, then the set $\{Mq + l \mid q \in \mathbb{Z}^m,\ l \in \mathbb{R}^n\}$ is dense in $\mathbb{R}^n$, where the matrix $M \in \mathbb{R}^{n \times m}$ is assembled with vectors $\alpha^{(i)}$, *i.e.* $M = [\alpha^{(1)}, \alpha^{(2)}, \cdots, \alpha^{(m)}]$. In the case of $m = 1$ and $n = 1$, let $\alpha = \sqrt{2}$, then Lemma 13 implies that the set $\{q\sqrt{2} \pm l \mid l \in \mathbb{N}^+,\ q \in \mathbb{N}^+\}$ is dense in $\mathbb{R}$. We will build upon this result to prove one of the most significant theorems in this article.

E Proofs of Section 4

In this appendix, we lay the groundwork for the proof of Theorem 8 by first introducing Lemma 14. We then present Theorem 8 and provide its complete proof, demonstrating that $\mathcal{T}^{\sigma}_{*,\mathcal{P}}$ can realize the UAP. To facilitate understanding of Theorem 8, we provide a simple illustrative example. While the theorem assumes dense positional encodings, we relax this condition under specific activation functions, as formalized in Lemma 15 and Theorem 9.

E.1 Lemma 14

Lemma 14. *For a network with a fixed width and a continuous activation function, it is possible to apply slight perturbations within an arbitrarily small error margin. For any network $\mathrm{N}_1^{\sigma}(x)$ defined on a compact set $\mathcal{K} \subset \mathbb{R}^{d_x}$, with parameters $A \in \mathbb{R}^{d_y \times k}, W \in \mathbb{R}^{k \times d_x}, b \in \mathbb{R}^{k \times 1}$, there exists $M > 0, M_1 > 0$ ($\|x\| < M$ and $\|a_i\| < M_1, i = 1, \cdots, k$), and for any $\varepsilon > 0$, there exists $0 < \delta < \frac{\varepsilon}{2M_1 k}$ and a perturbed network $\mathrm{N}_2^{\sigma}(x)$ with parameters $\tilde{A} \in \mathbb{R}^{d_y \times k}, \tilde{W} \in \mathbb{R}^{k \times d_x}, \tilde{b} \in \mathbb{R}^{k \times 1}$ ($\left\| \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| < M_1, i = 1, \cdots, k$), such that if $\max\{\|a_i - \tilde{a}_i\|, M\|w_i - \tilde{w}_i\| + \|b - \tilde{b}\| \mid i = 1, \cdots, k\} < \delta$, then*

$$\|N_1^{\sigma}(x) - N_2^{\sigma}(x)\| < \varepsilon, \quad \forall x \in \mathcal{K}, \tag{54}$$

where $a_i, \tilde{a}_i$ are the i-th column vectors of $A, \tilde{A}$, respectively, $w_i, \tilde{w}_i$ are the i-th row vectors of $W, \tilde{W}$, and $b_i, \tilde{b}_i$ are the i-th components of $b, \tilde{b}$, respectively, for any $i = 1, \cdots, k$.

Proof. We have $\mathrm{N}_1^{\sigma}(x) = \sum_{i=1}^{k} a_i \sigma(w_i x + b_i)$, where $a_i \in \mathbb{R}^{d_y}, w_i \in \mathbb{R}^{d_x}, b_i \in \mathbb{R}$, and $\tilde{\mathrm{N}}_2^{\sigma}(x) = \sum_{i=1}^{k} \tilde{a}_i \sigma(\tilde{w}_i x + \tilde{b}_i)$, where $\tilde{a}_j \in \mathbb{R}^{d_y}, \tilde{w}_i \in \mathbb{R}^{d_x}, \tilde{b}_i \in \mathbb{R}$. For any $x \in \mathcal{K}, \|x\| < M$. There exists a constant $M_1 > 0$ such that for any $i = 1, \cdots, k$, the following inequalities hold: $\|a_i\| < M_1$ and $\left\| \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| < M_1$.

Due to the continuity of the activation function, for any $\varepsilon > 0$, there exists $0 < \delta < \frac{\varepsilon}{2M_1 k}$, such that if $\|w_i x + b_i - (\tilde{w}_i x + \tilde{b}_i)\| \le \|w_i - \tilde{w}_i\|\|x\| + \|b_i - \tilde{b}_i\| < M\|w_i - \tilde{w}_i\| + \|b - \tilde{b}\| < \delta, i = 1, \cdots, k,$ then $\|\sigma(w_i x + b_i) - \sigma(\tilde{w}_i x + \tilde{b}_i)\| < \frac{\varepsilon}{2M_1 k}, i = 1, \cdots, k$, and $\|a_i - \tilde{a}_i\| < \delta, i = 1, \cdots, k$.

Combining all these inequalities, we can further derive:

$$
\begin{aligned}
&\|\mathrm{N}_1^\sigma(x) - \mathrm{N}_2^\sigma(x)\| \left\| \sum_{i=1}^k a_i \sigma(w_i x + b_i) - \sum_{i=1}^k \tilde{a}_i \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| \\
&\le \left\| \sum_{i=1}^k a_i \sigma(w_i x + b_i) - \sum_{i=1}^k a_i \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| + \left\| \sum_{i=1}^k a_i \sigma(\tilde{w}_i x + \tilde{b}_i) - \sum_{i=1}^k \tilde{a}_i \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| \\
&\le \max_i \|a_i\| \left\| \sum_{i=1}^k \sigma(w_i x + b_i) - \sum_{i=1}^k \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| + \max_i \left\| \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| \left\| \sum_{i=1}^k a_i - \sum_{i=1}^k \tilde{a}_i \right\|. \\
&\le \max_i \|a_i\| \sum_{i=1}^k \left\| \sigma(w_i x + b_i) - \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| + \max_i \left\| \sigma(\tilde{w}_i x + \tilde{b}_i) \right\| \sum_{i=1}^k \|a_i - \tilde{a}_i\| \\
&< M_1 k \frac{\varepsilon}{2M_1 k} + M_1 k \frac{\varepsilon}{2M_1 k} = \varepsilon
\end{aligned}
\tag{55}
$$

The proof is complete. □

E.2 Proof of Theorem 8

Theorem 8. *Let $\mathcal{T}^\sigma_{*,\mathcal{P}}$ be the class of functions $\mathrm{T}^\sigma_{*,\mathcal{P}}$ satisfying Assumption 1, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function σ, the subscript refers the finite vocabulary $\mathcal{V} = \mathcal{V}_x \times \mathcal{V}_y$, $\mathcal{P} = \mathcal{P}_x \times \mathcal{P}_y$ represents the positional encoding map, and denote a set S as:*

$$
S := \mathcal{V}_x + \mathcal{P}_x = \left\{ x_i + \mathcal{P}_x^{(j)} \;\middle|\; x_i \in \mathcal{V}_x,\ i,\ j \in \mathbb{N}^+ \right\}. \tag{56}
$$

If S is dense in $\mathbb{R}^{d_x}$, $\{1,\ -1,\ \sqrt{2},\ 0\}^{d_y} \subset \mathcal{V}_y$ and $\mathcal{P}_y = 0$, then $\mathcal{T}^\sigma_{*,\mathcal{P}}$ can achieve the UAP. More specifically, given a network $\mathrm{T}^\sigma_{*,\mathcal{P}}$, then for any continuous function $f : \mathbb{R}^{d_x - 1} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$ and $\varepsilon > 0$, there always exist $X \in \mathbb{R}^{d_x \times n}$ and $Y \in \mathbb{R}^{d_y \times n}$ from the vocabulary $\mathcal{V}$, i.e. *$x^{(i)} \in \mathcal{V}_x, y^{(i)} \in \mathcal{V}_y$, with some length $n \in \mathbb{N}^+$ such that*

$$
\left\| \mathrm{T}^\sigma_{*,\mathcal{P}}\left(\tilde{x}; X, Y\right) - f(x) \right\| < \varepsilon, \quad \forall x \in \mathcal{K}. \tag{57}
$$

Proof. Our conclusion holds for all element-wise continuous activation functions in $\mathcal{T}^\sigma_{*,\mathcal{P}}$. We now assume $d_y = 1$ for simplicity, and the case $d_y \neq 1$ will be considered later.

We are reformulating the problem. Using Lemma 3, we have,

$$
\mathrm{T}^\sigma_{*,\mathcal{P}}\left(\tilde{x}; X, Y\right) = U Y_{\mathcal{P}}\, \sigma\left((X + \mathcal{P})^\top B^\top C \tilde{x} \right) = U Y_{\mathcal{P}}\, \sigma\left(X_{\mathcal{P}}^\top B^\top C \tilde{x} \right). \tag{58}
$$

Since $\mathcal{P}_y = 0$, it follows that $Y_{\mathcal{P}} = Y$. For any continuous function $f : \mathbb{R}^{d_x - 1} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$ and for any $\varepsilon > 0$, we aim to show that there exists $\mathrm{T}^\sigma_{*,\mathcal{P}} \in \mathcal{T}^\sigma_{*,\mathcal{P}}$ such that:

$$
\begin{aligned}
&\left\| \mathrm{T}^\sigma_{*,\mathcal{P}}\left(\begin{bmatrix} x \\ 1 \end{bmatrix}; X, Y \right) - U f(x) \right\| < \|U\| \varepsilon, \quad \forall x \in \mathcal{K}, \\
\Leftrightarrow &\left\| Y\, \sigma\left(X_{\mathcal{P}}^\top B^\top C \tilde{x} \right) - f(x) \right\| < \varepsilon, \quad \forall x \in \mathcal{K}.
\end{aligned}
\tag{59}
$$

In the main text, for illustrative purposes, we consider the special case where U is the identity matrix to simplify the exposition. In the present analysis, we dispense with this assumption. We already have a Lemma 2 ensuring the existence of a one-hidden-layer network N^σ (with activation function

σ satisfying the required conditions) that approximates $f(x)$. Our proof is divided into four steps, serving as a bridge built upon the Lemma 2:

$$Y\,\sigma\left(X_{\mathcal{P}}^{\top}B^{\top}C\tilde{x}\right) \xrightarrow{\text{Lemma 2}} \mathrm{N}_*^{\sigma}(x) \xrightarrow{\text{step (3)}} \mathrm{N}'(x) \xrightarrow{\text{step (2)}} \mathrm{N}^{\sigma}(x) \xrightarrow{\text{step (1)}} f(x). \tag{60}$$

We present the specific details at each step.

Step (1): Approximating $f(x)$ Using $\mathrm{N}^{\sigma}(x)$. Supported by Lemma 2, there exists a neural network $\mathrm{N}^{\sigma}(x) = A\,\sigma(Wx+b) = \sum_{i=1}^{k} a_i\,\sigma(w_i x + b_i) \in \mathcal{N}^{\sigma}$, with parameters $k \in \mathbb{N}^{+}$, $A \in \mathbb{R}^{d_y \times k}$, $b \in \mathbb{R}^{k}$, and $W \in \mathbb{R}^{k\times(d_x-1)}$,

$$\|A\,\sigma(Wx+b) - f(x)\| < \frac{\varepsilon}{3}, \quad \forall x \in \mathcal{K}. \tag{61}$$

Step (2): Approximating $\mathrm{N}^{\sigma}(x)$ Using $\mathrm{N}'(x)$. Using Lemma 13 and Lemma 14, a neural network $\mathrm{N}^{\sigma}(x) = \sum_{i=1}^{k} a_i \sigma(w_i x + b_i) \in \mathcal{N}^{\sigma}$ can be perturbed into $\mathrm{N}'(x) = \sum_{i=1}^{k} (q\sqrt{2} \pm l)_i\,\sigma(\tilde{w}_i x + \tilde{b}_i)$ (with $q_i \in \mathbb{N}^{+}$ and $l_i \in \mathbb{N}^{+}, i = 1, \cdots, k$), such that for any $\varepsilon > 0$, there exists $0 < \delta < \frac{\varepsilon}{6M_1 k}$ satisfying:

$$\max\{\|a_i - (q\sqrt{2} \pm l)_i\|, M\|w_i - \tilde{w}_i\| + \|b - \tilde{b}\| \mid i = 1, \cdots, k\} < \delta, \tag{62}$$

ensuring:

$$\|\mathrm{N}^{\sigma}(x) - \mathrm{N}'(x)\| = \left\| \sum_{i=1}^{k} a_i\,\sigma(w_i x + b_i) - \sum_{i=1}^{k} (q\sqrt{2} \pm l)_i\,\sigma(\tilde{w}_i x + \tilde{b}_i) \right\| < \frac{\varepsilon}{3}, \quad \forall x \in \mathcal{K}. \tag{63}$$

Step (3): Approximating $\mathrm{N}'(x)$ Using $\mathrm{N}_*^{\sigma}(x)$. Next, we show that $\mathrm{N}_*^{\sigma}(x) = \sum_{i=1}^{n} y^{(i)}\,\sigma(\tilde{R}_i \tilde{x}) \in \mathcal{N}_*^{\sigma}$ can approximate $\mathrm{N}'(x) = \sum_{i=1}^{k} (q\sqrt{2} \pm l)_i\,\sigma(\tilde{w}_i \tilde{x})$. As a demonstration, we approximate a single term $(q\sqrt{2} \pm l)_1\,\sigma(\tilde{w}_1 \tilde{x})$. Since the positional encoding is fixed, *i.e.* $\mathcal{V}_x + \mathcal{P}^{(1)}$ is a finite set, one of two cases must occur:

1. *Valid Position:* If there exists $x^{(1)} \in \mathcal{V}_x$ where $(x^{(1)} + \mathcal{P}^{(1)})^{\top} B^{\top} C \approx \tilde{w}_1$
2. *Invalid Position:* Set $y^{(1)} = 0$ to nullify contribution

Since S is dense in $\mathbb{R}^{d_x}$ and $B^{\top}C$ is non-singular, the set $G := \{\tilde{R} \mid \tilde{R} = X_{\mathcal{P}}^{\top} B^{\top} C, X_{\mathcal{P}} \subset 2^{S}\}$ remains dense. Let K_1 denote the set of indices corresponding to all "valid" positions for $\tilde{w}_1$. Since $y^{(i)} \in \{1, -1, \sqrt{2}, 0\}$, we require $q_1 + l_1$ elements from G that approximate $\tilde{w}_1$, such that

$$\begin{aligned}
&\left\| \sum_{j \in K_1} y^{(j)}\,\sigma(\tilde{R}_j \tilde{x}) - (q\sqrt{2} \pm l)_1\,\sigma(\tilde{w}_1 \tilde{x}) \right\| \\
&= \|\sqrt{2} \sum_{j \in Q_1} \sigma(\tilde{R}_j \tilde{x}) \pm \sum_{j \in L_1} \sigma(\tilde{R}_j \tilde{x}) - (q\sqrt{2} \pm l)_1\,\sigma(\tilde{w}_1 \tilde{x})\| \\
&< \frac{\varepsilon}{3k}, \quad \forall x \in \mathcal{K}.
\end{aligned} \tag{64}$$

Here, $\#(K_1) = q_1 + l_1$ and $K_1 = Q_1 \bigcup L_1$, where Q_1, L_1 are disjoint subsets of positive integer indices satisfying $\#(Q_1) = q_1$ and $\#(L_1) = l_1$. For this construction, we assign $y^{(j)} = \sqrt{2}$ for $j \in Q_1$ and $y^{(j)} = \pm 1$ for $j \in L_1$. For $j \in \{1, 2, 3, \cdots, \max_i\{i \in K_1\}\} \backslash K_1$, *i.e.*, for the *Invalid Position*, we set $y^{(j)} = 0$.

The multi-term approximation employs parallel construction via disjoint node subsets $K_i = Q_i \cup L_i$, where Q_i (q_i nodes) and L_i (l_i nodes) implement $\sqrt{2}$ and ± 1 coefficients respectively. For $j \notin \bigcup_{l=1}^{k} K_l$, we set $y^{(j)} = 0$. Each term achieves:

$$\left\| \sum_{j \in K_i} y^{(j)} \sigma(\tilde{R}_j \tilde{x}) - (q\sqrt{2} \pm l)_i \sigma(\tilde{w}_i \tilde{x}) \right\| < \frac{\varepsilon}{3k}. \tag{65}$$

We then define $n = \max\{j \mid j \in \bigcup_{l=1}^{k} K_l\}$. The complete network combines these approximations through:

$$\|\mathrm{N}_*^\sigma(x) - \mathrm{N}'(x)\| = \left\| \sum_{i=1}^{n} y^{(i)}\, \sigma(\tilde{R}_i \tilde{x}) - \sum_{i=1}^{k} (q\sqrt{2} \pm l)_i\, \sigma(\tilde{w}_i \tilde{x}) \right\| < \frac{\varepsilon}{3}, \quad \forall x \in \mathcal{K}. \tag{66}$$

Step (4): Combining Results. Combining all results, we have:

$$\begin{aligned} \|Y\, \sigma\left(X_{\mathcal{P}}^\top B^\top C \tilde{x}\right) - f(x)\| &= \|\mathrm{N}_*^\sigma(x) - f(x)\| \\ &< \|\mathrm{N}_*^\sigma(x) - \mathrm{N}'(x)\| + \|\mathrm{N}'(x) - \mathrm{N}^\sigma(x)\| + \|\mathrm{N}^\sigma(x) - f(x)\| \\ &< \varepsilon, \quad \forall x \in \mathcal{K}. \end{aligned} \tag{67}$$

The scalar-output results ($d_y = 1$) extend naturally to vector-valued functions via component-wise approximation. For any continuous $f : \mathbb{R}^{d_x - 1} \to \mathbb{R}^{d_y}$ on a compact domain $\mathcal{K}$, uniform approximation is achieved by independently approximating each coordinate function f_j with scalar networks $\mathrm{N}_{*,j}^\sigma(x)$ satisfying

$$\left\|\mathrm{N}_{*,j}^\sigma(x) - f_j(x)\right\| < \frac{\varepsilon}{\sqrt{d_y}}, \quad \forall x \in \mathcal{K}. \tag{68}$$

The full approximator is then obtained by concatenating the component networks.

$$\mathrm{N}_*^\sigma(x) = \begin{bmatrix} \mathrm{N}_{*,1}^\sigma(x) \\ \vdots \\ \mathrm{N}_{*,d_y}^\sigma(x) \end{bmatrix}, \quad \|\mathrm{N}_*^\sigma(x) - f(x)\| < \varepsilon, \tag{69}$$

$$\mathrm{N}_{*,j}^\sigma(x) = \sum_{i=1}^{n} y_j^{(i)}\, \sigma(\tilde{R}_i \tilde{x}), \tag{70}$$

where $y_j^{(i)}$ is the j-th row of the $y^{(i)}$. We require that the index sets satisfy $K_i^{(o)} \cap K_j^{(u)} = \emptyset$ for all $o, u, i, j \in \mathbb{N}^+$, where $K_i^{(o)}$ denotes the index set constructed for the i-th term approximation in the o-th output dimension. Furthermore, each $y^{(j)}$ must have at most one non-zero element across its dimensions. This ensures we achieve uniform approximation by independently handling each output dimension. The proof is complete. □

E.3 Example of Theorem 8

We present a concrete example with 2D input ($d_x = 2$) and 2D output ($d_y = 2$) to illustrate the universal approximation capability of our architecture. Consider a continuous function $f : [0,1]^2 \to \mathbb{R}^2$ defined by

$$f(x_1, x_2) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix}. \tag{71}$$

Our goal is to construct a module $\mathrm{T}_{*,\mathcal{P}}^\sigma$ such that

$$\left\| \mathrm{T}_{*,\mathcal{P}}^\sigma \left(\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}; X, Y \right) - f(x_1, x_2) \right\| < \varepsilon. \tag{72}$$

Step (1): Component-wise Approximation. For each component f_i, there exists a single-hidden-layer neural network $\mathrm{N}_i^\sigma(x) = A_i \sigma(W_i x + b_i)$ such that

$$\sup_{x \in [0,1]^2} \|f_i(x) - \mathrm{N}_i^\sigma(x)\| < \frac{\varepsilon}{6\sqrt{2}}, \quad i = 1, 2. \tag{73}$$

Step (2): Rational Perturbation. We approximate each N_i^σ by a rational network N_i':

$$\mathrm{N}_1'(x) = (3\sqrt{2} - 2)\sigma(\tilde{w}_1^\top \tilde{x}), \tag{74}$$

$$\mathrm{N}_2'(x) = (2\sqrt{2} + 1)\sigma(\tilde{w}_2^\top \tilde{x}), \tag{75}$$

where $\tilde{x} = [x_1 \quad x_2 \quad 1]^\top$, satisfying

$$\sup_{x\in[0,1]^2} \|\mathrm{N}_i^\sigma(x) - \mathrm{N}_i'(x)\| < \frac{\varepsilon}{6\sqrt{2}}, \quad i = 1, 2. \tag{76}$$

Step (3): Architecture Realization. We define a Transformer-like module $\mathrm{N}_*^\sigma(x)$ with shared representation:

$$\tilde{R} = [\approx \tilde{w}_1 \quad \approx \tilde{w}_1 \quad \approx \tilde{w}_1 \quad \approx \tilde{w}_1 \quad \approx \tilde{w}_1 \quad \approx \tilde{w}_2 \quad \approx \tilde{w}_2 \quad \approx \tilde{w}_2]^\top, \tag{77}$$

$$Y = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & 1 \end{bmatrix}, \tag{78}$$

such that

$$\mathrm{N}_*^\sigma(x) = \begin{bmatrix} \sum_{i=1}^8 y_1^{(i)} \sigma(\tilde{R}_i^\top \tilde{x}) \\ \sum_{i=1}^8 y_2^{(i)} \sigma(\tilde{R}_i^\top \tilde{x}) \end{bmatrix}, \quad \sup_{x\in[0,1]^2} \|\mathrm{N}_i'(x) - \mathrm{N}_{*,i}^\sigma(x)\| < \frac{\varepsilon}{6\sqrt{2}}. \tag{79}$$

Step (4): Error Analysis. The total approximation error satisfies

$$\|f(x) - \mathrm{N}_*^\sigma(x)\| \leq \sqrt{\sum_{i=1}^{2} \left(\|f_i - \mathrm{N}_i^\sigma\| + \|\mathrm{N}_i^\sigma - \mathrm{N}_i'\| + \|\mathrm{N}_i' - \mathrm{N}_{*,i}^\sigma\|\right)^2} \tag{80}$$

$$\leq \sqrt{2 \cdot \left(\frac{\varepsilon}{2\sqrt{2}}\right)^2} = \frac{\varepsilon}{2} < \varepsilon. \tag{81}$$

Implementation Details. Node allocation is shown in Table 2.

Table 2: Node allocation for 2D output example

Node Index	$y^{(i)}$	$\tilde{R}_i$	Purpose
1–3	$(\sqrt{2}, 0)$	$\approx \tilde{w}_1$	$3\sqrt{2}$ term for $\sigma(\tilde{w}_1^\top \tilde{x})$
4–5	$(-1, 0)$	$\approx \tilde{w}_1$	-2 term for $\sigma(\tilde{w}_1^\top \tilde{x})$
6–7	$(0, \sqrt{2})$	$\approx \tilde{w}_2$	$2\sqrt{2}$ term for $\sigma(\tilde{w}_2^\top \tilde{x})$
8	$(0, 1)$	$\approx \tilde{w}_2$	1 term for $\sigma(\tilde{w}_2^\top \tilde{x})$

Alternative Construction. A compact design uses:

$$Y = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ -1 & -1 & 0 & 1 & 0 \end{bmatrix}, \quad \tilde{R} = \begin{bmatrix} \approx \tilde{w}_1 \\ \approx \tilde{w}_1 \\ \approx \tilde{w}_1 \\ \approx \tilde{w}_2 \\ \approx \tilde{w}_2 \end{bmatrix}, \tag{82}$$

which reduces the number of tokens but complicates the analysis in high dimensions. We thus adopt disjoint index sets to ensure analytical tractability.

E.4 Proof of Theorem 9

Before prove Theorem 9, we need to prove the following lemma with the help of the well-known Stone-Weierstrass theorem.

Lemma 15. *For any continuous function $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ defined on a compact domain $\mathcal{K}$, and for any $\varepsilon > 0$, there exist a network $\mathrm{N}^{\exp}(x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ satisfying*

$$\|\mathrm{N}^{\exp}(x) - f(x)\| < \varepsilon, \quad \forall x \in \mathcal{K}, \tag{83}$$

where $b = 0$ and all row vectors of W are restricted in a neighborhood $B(\omega^*, \delta)$ with any prefixed $w^* \in \mathbb{R}^{d_x}$ and radius $\delta > 0$.

Proof. Assume $f(x) = (f_1(x), \cdots, f_{d_x}(x))$. According to Stone-Weierstrass theorem, for any $\varepsilon > 0$, there exist polynomials $P_i(x)$ satisfying

$$\begin{aligned} &\max_{x\in\mathcal{K}} \|P_i(x) - f_i(x)\mathrm{e}^{-w^*\cdot x}\| < \frac{\varepsilon}{2\max_{x\in\mathcal{K}}\|\mathrm{e}^{w^*\cdot x}\|}, \\ &\Rightarrow \max_{x\in\mathcal{K}} \|P_i(x)\mathrm{e}^{w^*\cdot x} - f_i(x)\| < \frac{\varepsilon}{2}, \quad i = 1, 2, \cdots, d_x. \end{aligned} \tag{84}$$

Then we construct a single-layer FNN with exponential activation function to approximate $P_i(x)\mathrm{e}^{w^*\cdot x}$. The multiple derivatives of $h(w) := \mathrm{e}^{w\cdot x} = \exp(w_1x_1 + \cdots + w_{d_x}x_{d_x})$ with respect to $w_1, \cdots, w_{d_x}$ is

$$\frac{\partial^{|\alpha|}h}{\partial w^\alpha} = \frac{\partial^{|\alpha|}h}{\partial w_1^{\alpha_1}\cdots\partial w_{d_x}^{\alpha_{d_x}}}, \tag{85}$$

where $\alpha \in \mathbb{N}^{d_x}$ represents the index and $|\alpha| := \alpha_1 + \cdots + \alpha_{d_x}$. Actually, the form of multiple derivative $\frac{\partial^{|\alpha|}h}{\partial w^\alpha}$ is a polynomial of $|\alpha|$ degree with respect to $x_1, \cdots, x_{d_x}$ times $h(w)$. Hence, each target term $P_i(x)\mathrm{e}^{w^*\cdot x}$ can be written as a linear combination of such multiple derivatives of $h(w)$, which allow us to approximate the required partials and thus complete the proof. And multiple derivative can be approximated by finite difference method, and the approach of finite difference method can be done by one hidden layer. □

Remark 16. *We give two examples of approximating multiple derivatives of $h(w)$ below.*

$$\begin{aligned} x_1h(w) &= \frac{\partial h}{\partial w_1}\Big|_{w=w^*} \\ &= \frac{h(w^*+\lambda e_1) - h(w^*)}{\lambda} + R_1(\lambda, w^*) \\ &= \lambda^{-1}h(w^*+\lambda e_1) - \lambda^{-1}h(w^*) + R_1(\lambda, w^*), \end{aligned} \tag{86}$$

and

$$\begin{aligned} x_1x_2h(w) &= \frac{\partial^2 h}{\partial w_1\partial w_2}\Big|_{w=w^*} \\ &= \frac{h(w^*+\lambda(e_1+e_2)) - h(w^*+\lambda e_1) - h(w^*+\lambda e_2) + h(w^*)}{\lambda} + R_2(\lambda, w^*) \\ &= \lambda^{-1}h((w^*+\lambda(e_1+e_2))\cdot x) - \lambda^{-1}h((w^*+\lambda e_1)\cdot x) - \\ &\quad \lambda^{-1}h((w^*+\lambda e_2)\cdot x) + \lambda^{-1}h(w^*\cdot x) + R_2(\lambda, w^*), \end{aligned} \tag{87}$$

where $e_1 = (1, 0, 0, \cdots, 0),\ e_2 = (0, 1, 0, \cdots, 0)$ are unite vectors and $R_1(\lambda, w^*),\ R_2(\lambda, w^*)$ are error terms with respect to λ and w^*. According to Taylor's theorem, the error terms $R_1(\lambda, w^*) = \lambda\frac{\partial^2 h}{\partial w_1^2}\big|_{w=\xi}$ for some ξ between w^* and $w^* + \lambda e_1$. It is obvious that the partial differential term is uniformly bounded, so the resulting error can be made arbitrarily small by a suitable choice of the parameter λ. The argument for $R_2(\lambda, W^*)$is entirely analogous and is therefore omitted; see [60] for further details.

Since λ is very small and the exponential term $\mathrm{e}^{w^*\cdot x}$ only involves the parameters $w^*,\ w^* + e_1$ and $w^* + e_2$, which all lie within a small neighborhood of w^*, the desired conclusion can be drawn, and this means we can actually restrict that all row vectors of W are restructed in $B(W, \delta)$.

Theorem 9 (Formal Version)**.** *Let $\mathcal{T}^\sigma_{*,\mathcal{P}}$ be the class of functions $\mathrm{T}^\sigma_{*,\mathcal{P}}$ satisfying Assumption 1, with a non-polynomial, locally bounded, piecewise continuous element-wise activation function σ, the* *subscript refers the finite vocabulary $\mathcal{V} = \mathcal{V}_x \times \mathcal{V}_y$, $\mathcal{P} = \mathcal{P}_x \times \mathcal{P}_y$ represents the positional encoding map, and denote a set S as:*

$$S := \mathcal{V}_x + \mathcal{P}_x = \left\{ x_i + \mathcal{P}_x^{(j)} \;\middle|\; x_i \in \mathcal{V}_x,\ i,\ j \in \mathbb{N}^+ \right\}. \tag{88}$$

If the set S is dense in $[-1,1]^{d_x}$, then $\mathcal{T}_{*,\mathcal{P}}^{\mathrm{ReLU}}$ is capable of achieving the UAP. Additionally, if S is only dense in a neighborhood $B(w^*, \delta)$ of a point $w^* \in \mathbb{R}^{d_x}$ with radius $\delta > 0$, then the class of transformers with exponential activation, i.e. *$\mathcal{T}_{*,\mathcal{P}}^{\exp}$, is capable of achieving the UAP.*

Proof. For the proof of ReLU case, we follow the same reasoning as in the pervious one, noting the $\mathrm{ReLU}(ax) = a\,\mathrm{ReLU}(x)$ holds for any positive a. In the proof of Theorem 8, we construct a $\mathrm{T}_{*,\mathcal{P}}^{\mathrm{ReLU}}(\tilde{x}; X, Y) \in \mathcal{T}_{*,\mathcal{P}}^{\mathrm{ReLU}}$ to approximate a FNN $A\,\mathrm{ReLU}(Wx + B)$. Here we can do the similar construction to find another $\tilde{\mathrm{T}}_{*,\mathcal{P}}^{\mathrm{ReLU}}(\tilde{x}; X, Y) \in \mathcal{T}_{*,\mathcal{P}}^{\mathrm{ReLU}}$ to approximate $\lambda A\,\mathrm{ReLU}\left(\lambda^{-1}(Wx + b)\right)$ as the second to the forth steps in Theorem 8, where λ is big enough to make the row vectors in $\lambda^{-1}W$ is small enough so that $S = \{x_i + \mathcal{P} \mid x_i \in \mathcal{V},\ i,\ j \in \mathbb{N}^+\}$ is dense in $[-1,1]^{d_x}$ is sufficient.

For exponential Transformers, by using Lemma 15, we can do the second step to the forth steps in Theorem 8 again, which is similat to ReLU case. □

F Weakened Assumption and Generalized Conclusions

It is important to note that most of out conclusions remain valid even if Assumption 1 is weakened. Below we outline the reasoning.

In general, we decompose the matrices as follows:

$$Q^\top K = \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}, V = \begin{bmatrix} D & E \\ F & U \end{bmatrix}, \tag{89}$$

where $O_{11},\ D \in \mathbb{R}^{d_x \times d_x}$, $O_{12},\ E \in \mathbb{R}^{d_x \times d_y}$, $O_{21},\ F \in \mathbb{R}^{d_y \times d_x}$, and $O_{22},\ U \in \mathbb{R}^{d_y \times d_y}$, respectively. The attention mechanism can then be computed as:

$$\begin{aligned}
\mathrm{Attn}_{Q,K,V}^{\sigma}(Z) &= VZM\sigma(Z^\top Q^\top KZ) \\
&= \begin{bmatrix} D & E \\ F & U \end{bmatrix} \begin{bmatrix} X & x \\ Y & 0 \end{bmatrix} \begin{bmatrix} I_n & \\ & 0 \end{bmatrix} \sigma\left(\begin{bmatrix} X^\top & Y^\top \\ x^\top & 0 \end{bmatrix} \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} \begin{bmatrix} X & x \\ Y & 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} DX + EY & 0 \\ FX + UY & 0 \end{bmatrix} \sigma\left(\begin{bmatrix} O & \left(X^\top O_{11} + Y^\top O_{21}\right)x \\ x^\top\left(O_{11}X + O_{12}Y\right) & x^\top O_{11} x \end{bmatrix} \right),
\end{aligned} \tag{90}$$

where O represents the matrix $X^\top O_{11} X + X^\top O_{12} Y + Y^\top O_{21} X + Y^\top O_{22} Y$. As a result, we have:

$$\mathrm{T}^{\sigma}\left(\tilde{x}; X, Y\right) = (FX + UY)\sigma\left(\left(X^\top O_{11} + Y^\top O_{21}\right)\tilde{x} \right), \tag{91}$$

for the case of element-wise activations, and:

$$\mathrm{T}^{\mathrm{softmax}}(\tilde{x}; X, Y) = (FX + UY)\left(\mathrm{softmax}\left(\begin{bmatrix} (X^\top O_{11} + Y^\top O_{21})\tilde{x} \\ \tilde{x}^\top O_{11}\tilde{x} \end{bmatrix} \right) \right)_{1:n}, \tag{92}$$

for the case of softmax activation.

By revisiting the definition of T^{σ} and T_*^{σ}, and comparing T^{σ} presented here with those in the preceding section, it is clear that the only distinction lies in the specific matrices involved, and matrix O_{11} and U is non-singular are the only conditions we need. Notably, the proof process for Theorem 7 does not rely on any assumption, which means this conclusion stated in Section 3 can be further strengthened.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes] .

   Justification: We outline and discuss the contributions and scope of our work at the end of the Abstract and in a dedicated subsection 1.1 of the Introduction 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes] .

   Justification: We discuss the limitations of our work around line 345 in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes] .

Justification: Our assumptions are clearly stated as Assumption 1 in Section 2 and Appendix F, and the proofs are provided in Appendix B to Appendix E.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [NA] .

Justification: This paper does not include experiments.

Guidelines:

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

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

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

Answer: [NA] .

Justification: This paper does not include experiments requiring code.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [NA] .

   Justification: This paper does not include experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [NA] .

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [NA] .

   Justification: This paper does not include experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes] .

   Justification: This research is theoretical in nature and does not involve human subjects, personal data, or potentially harmful applications. All results are derived through mathematical analysis and do not raise ethical concerns as outlined in the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: This paper focuses on the theoretical expressivity of Transformers under ICL and provides approximation results from a mathematical perspective. We believe that discussing societal impact falls outside the scope of this foundational contribution.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper does not release any data, models, or tools that pose risks of misuse or dual use. The work is purely theoretical and focuses on the UAP in VICL with single-layer Transformers.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [NA]

    Justification: The paper does not use any external assets such as datasets, models, or third-party code. The research is purely theoretical and does not rely on pre-existing software or data resources.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: This paper is theoretical in nature and does not introduce or release any new datasets, models, or software assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This paper does not involve any crowdsourcing or experiments with human subjects.

    Guidelines:

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

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

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

    Answer: [NA] .

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA] .

    Justification: The core method development in this research does not involve LLMs

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.