UNIVERSAL CONSISTENCY OF TRANSFORMERS AS FUNCTIONAL REGRESSORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers are an indispensable component of modern foundation models, yet their theoretical understanding for functional regression on non-Euclidean domains remains underexplored. While the universal approximation properties of Euclidean Transformers have been extensively studied [62], their statistical consistency as functional regressors remains less understood, particularly beyond Euclidean settings. Motivated by recent advances in hyperbolic neural networks and the intrinsic hierarchies of data such as text, images, and graphs, we develop a statistical framework for analyzing Transformers in hyperbolic spaces. We establish universal approximation results for broad classes of sequence-to-sequence functional maps and prove universal statistical consistency for least-squares functional regression without imposing a parametric model specification. We further derive stochastic error bounds for the corresponding empirical estimators, yielding an asymptotic rate of $\mathcal{O}(t^{-1/2d})$, where t denotes the number of input tokens and d the input embedding dimension. Our analysis explicitly characterizes the curse of dimensionality induced by the ambient embedding dimension, providing a quantitative account of the statistical complexity of nonlinear attention-based Transformer architectures. The Euclidean setting is recovered as a special case of the proposed framework. Experiments on real datasets with both continuous and categorical responses empirically support our theoretical results and demonstrate the practical relevance of hyperbolic representations for functional sequence-to-sequence regression.

1 INTRODUCTION

Over the last decade, Transformers [57] have played a crucial role in influencing the domain of Natural Language Processing (NLP). Initially, Transformers were designed to solve sequence-to-sequence tasks like machine translation [58], language modeling [7], question-answering [49], etc. Inspired by the remarkable success of language Transformers, Vision Transformers [15] and Graph Transformers [16] are subsequently proposed to improve the performance of image- and graph-related tasks. Currently, Transformers are pivotal components in developing a wide set of Large Language Models (LLMs) [52, 54].

The impressive performances of Transformers are underpinned by the self-attention modules, which enable dynamic modeling of pairwise token interactions in the embedding space. Transformers comprise multiple encoder-decoder blocks, where encoders perform self-attention operations and decoders compute masked attention and cross-attention to model autoregressive dependencies. Both encoder and decoder contain a series of feedforward layers with residual connections. On top of that, positional encodings are incorporated to provide order-aware information to Transformers [39, 17].

Despite the surprising empirical success, their theoretical analysis remains comparatively underexplored. A plethora of research focuses on expressivity that establishes connections between Transformers and formal languages, with Transformers and tighter bounds for encoders [9, 3]. Some works explicitly show the relation between the Transformer and Boolean circuits [25, 38] and Turing completeness of the self-attention modules [45, 6]. These investigations shed light on the syntactic capabilities of Transformers, but often leave open the question of their statistical learning guarantees.

One significant work [62] establishes that Transformers with two attention heads (each with a size of one) and a feed-forward layer with a hidden dimension of four can universally approximate any

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

compactly supported sequence-to-sequence function. Despite its long-standing impact, **the universal approximation alone can not suffice to understand the generalization power of Transformers**; **particularly the almost sure convergence of the excess risk to** 0 **under a mis-specified set-up (referred to as *universal consistency* [23]), when the estimators are modeled as empirical squared** l^2 **loss minimizers of a functional regression problem**. Another work [30] proves that Transformers that comprise only a self-attention layer with low-rank weight matrices are a universal approximator of equivariant functions on a compact set. Furthermore, [59] shows that feedforward networks can be approximated with Boolean circuits, and Transformers can be statistically meaningfully approximated with Turing machines. Beyond the universal approximation properties, analysing the consistency of the Transformer architecture can address the gap.

Recent works on in-context learning [2, 43, 21] consider the Transformer as a functional regressor. Apart from these works, [43, 31, 42] examine the capability of Transformers for solving linear regression. Our study generalizes to the full scope of nonlinear regression under the mean squared error (MSE) loss. This choice departs from the conventional cross-entropy objective [29, 53, 40], but aligns with our continuous output domain and the goal of estimating conditional expectations as a minimizer of the functional regression problem.

Universal consistency is defined in terms of Bayes risk [23], which is fundamentally rooted in the regression problem. Thus, considering Mean Squared Error (MSE) as the optimization objective links with the theoretical error minimization target. This approach ensures a statistically meaningful convergence analysis.

In this work, we present a unified view of Transformers by generalizing the notion of tokens across modalities. In language tasks, tokens correspond to words or phrases; in ViTs, image patches serve as tokens; and in graph data, nodes, edges, or subgraphs act as tokens. This abstraction allows us to prove the universal consistency of Transformers under a unified token formulation. **To the best of our knowledge, this is the first work to establish universal statistical consistency for Transformers and their variants without imposing explicit model-specific assumptions**, further supported by extensive experiments on language, image, and graph datasets, which align closely with our theoretical claims.

Contributions. Our main contributions are:
1. We characterize the generalization behavior of hyperbolic Transformers when employed as functional regressors and trained with the mean squared error (MSE) loss.
2.We prove universal consistency of Transformers under feasible training assumptions and provide a stochastic convergence rate bound of empirical error to Bayes error. Our analyses show that the consistency of the Euclidean Transformer can be recovered as a special case of the hyperbolic treatment.
3.Since most of the modern neural architectures take arbitrary sequence of vectors(tokens) as inputs, **our theoretical claims are seamlessly verifiable to language, vision, and graph Transformers through the unified token abstraction, without the need to modify it to a fundamentally new theoretical generality**.

Before proceeding, we emphasize that this work does not propose a new Transformer architecture. Rather, our primary objective is to establish universal statistical consistency for the Transformer architecture and its variants. To broaden the scope of our theoretical analysis, we develop the underlying proof techniques in a hyperbolic setting, motivated by recent advances in hyperbolic neural networks and their applications to non-Euclidean data. The Euclidean setting then emerges naturally as a special limiting case of our analysis, allowing the corresponding consistency results to be recovered within the same theoretical framework.

2 Related Works

Attention Mechanisms. Since the inception of self-attention [57], many variants have aimed to better capture contextual meaning in NLP. A key observation that the attention matrices are often sparse led to Sparse Attention Mechanisms [10], which reduce computation by ignoring unrelated tokens. Notable approaches include Star Transformer with band or global attention [46], LongFormer with inter-global attention [5], and BigBird with external global attention [63, 27]. Sparse Sinkhorn Attention [51] further introduces blockwise query-key partitioning with Sinkhorn normalization for locality-aware, doubly stochastic matrices.
Neural Networks and Universal Approximation. The universal approximation theorem [11, 28]

Table 1: The details of the notations are presented.

Notation	Description
$\mathbb{D}_c^d$	Poincaré Ball in dimension d with curvature $-c$
$T_p(\mathbb{D}_c^d)$	Tangent space at $p \in \mathbb{D}_c^d$
$\exp_x^c(v)$	Projects $v \in T_x(\mathbb{D}_c^d)$ to a point in $\mathbb{D}_c^d$ [Euclidean $\rightarrow$ Poincaré Ball]
$\log_y^c(x)$	Projects $x \in \mathbb{D}_c^d$ to a point in $T_y(\mathbb{D}_c^d)$ [Poincaré Ball$\rightarrow$ Euclidean]
$\oplus_c$	Möbius addition on Poincaré disc with curvature $-c$
$\otimes_c$	Möbius scalar multiplication on Poincaré disc with curvature $-c$
$t_{\mathcal{H}}^{h,s,r}$	Transformer block with h heads, head size s, dim r
$\mathcal{T}_{\mathcal{H}}^{h,s,r}$	Function class represented by $t_{\mathcal{H}}^{h,s,r}$ block
$t_{\mathcal{P},\mathcal{H}}^{h,s,r}$	Positionally encoded Transformer block
$\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$	Function class represented by $t_{\mathcal{P},\mathcal{H}}^{h,s,r}$ block
$\mathcal{E}_{\pi_M,D}(f)$	$\frac{1}{n}\sum_{i=n}^{t} \|f(X_i) - \pi_M\left(\log_0^c(Y_i)\right)\|^2$
$\mathcal{E}_{\pi_M}(f)$	$\int_{\mathcal{X}\times\mathcal{Y}} \|f(x) - \pi_M\left(\log_0^c(y)\right)\| \, d\rho$

established that single-hidden-layer networks can approximate any continuous function on compact domains. Later work examined depth [37, 35], with Hanin and Sellke [24] showing that a hidden layer of size $s + 1$ suffices for scalar-valued functions of s-dimensional inputs. Universality has also been extended to CNNs [64]. For sequence models, Transformers with positional encodings and residuals are universal approximators [62], and ReLU networks are universal for permutation-equivariant functions [48].

Hyperbolic Neural Networks and Transformers. Hyperbolic architectures exploit non-Euclidean geometry to model hierarchical and tree-like data more efficiently. Hyperbolic neural networks extend Euclidean layers with Riemannian optimization and Möbius operations [20, 41], and hyperbolic CNNs and GNNs further improve representation of hierarchies [8]. More recently, hyperbolic Transformers combine attention with hyperbolic geometry to capture long-range dependencies while preserving hierarchical structure [50, 22, 44], underscoring the promise of non-Euclidean deep learning.

3 Extending Framework to a Generalized Set-up

Recent advances in hyperbolic neural networks have demonstrated the benefits of geometry-aware representations for hierarchical and complex structured data. In particular, graph and network data often exhibit intrinsically non-Euclidean geometry that may be inefficiently represented in Euclidean spaces, requiring unnecessarily high-dimensional embeddings. This motivates extending the theoretical analysis of Transformers to hyperbolic spaces, where negative curvature can better capture such structures. We first illustrate this motivation through a concrete example, followed by a detailed mathematical formulation and generalization of the Transformer architecture to hyperbolic spaces.

3.1 A Motivating Example

We conduct a study on the MNIST dataset to observe the efficacy of the BERT model [13] enhanced with hyperbolic geometry. The encoder component of a standard BERT is re-designed to embed features on a Poincaré Ball model. Our encoder comprises 2 attention heads, each of size 1, and a feed-forward layer of hidden dimension 4. The experimental outcomes are demonstrated in Figure 1.

In Figure 1(a), we represent the test RMSE loss of the model w.r.t. the number of training samples for three curvature settings: $c = 0.0$ (Euclidean), $c = 1.0$, and $c = 10.0$. Each experiment is repeated over 5 training iterations per sample size, where the sample sizes varied from 5000 to 60000 with an increment of 5000. The comparative study illustrates that the hyperbolic variants exhibit a significant reduction in test RMSE loss, indicating improved generalization.

We also visualized embeddings in both Euclidean and hyperbolic spaces. Figure 1(b) depicts the 2-dimensional Euclidean embeddings ($\mathbb{R}^2$) produced by the standard BERT. Furthermore, Figures 1(c) and 1(d) represent the image embeddings in 2-d Poincaré ball ($\mathbb{D}_c^2$) with curvatures $c = 1.0$, and $c = 10.0$, respectively. A close observation reveals that the clusters formed in hyperbolic spaces are better separated and capture intricate patterns in the underlying data compared to their Euclidean

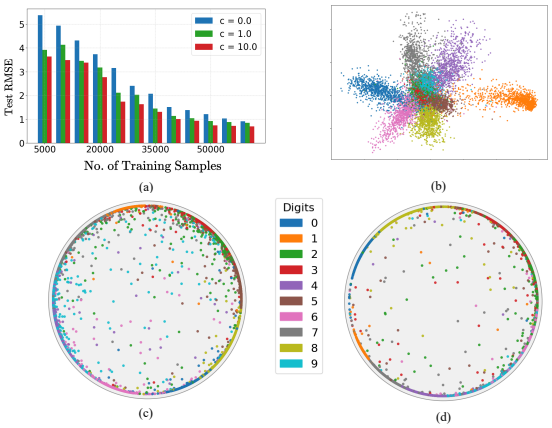


Figure 1: Hyperbolic BERT is applied to the MNIST dataset. (a) The test RMSE loss diminishes faster when the curvature increases. (b) The embeddings are also well-separated compared to their Euclidean counterparts. In (c) and (d), the image embeddings move closer to the boundary of the Poincaré ball as the curvature increases.

counterparts. Notably, the embeddings dispersed more toward the boundary of the ball with the increasing space curvatures.

In the remaining part of this section, we will first discuss the fundamentals of vanilla Transformers. Then, we will extend the existing framework to a generalized hyperbolic variant suitable for our theoretical analyses. All the necessary notations are described in Table 1.

3.2 Preliminaries on Vanilla Transformers

Consider a L-length sequence of input tokens $S = \{s_1, \ldots, s_L\}$ with token embeddings $\mathbf{X} = [X_1, \ldots, X_L] \in \mathbb{R}^{d \times L}$, where each $X_i \in \mathbb{R}^d$. Following [57], the Query, Key, and Value matrices are

$$\mathbf{Q} = \mathbf{X}^\top \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}^\top \mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}^\top \mathbf{W}_V \in \mathbb{R}^{L \times d_k}, \tag{1}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ are learnable projection matrices. The single-head self-attention operation is

$$\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\mathbf{Q}\mathbf{K}^\top\right)\mathbf{V} \in \mathbb{R}^{L \times d_k}. \tag{2}$$

For h attention heads, the multi-head self-attention (MSA) module concatenates the individual heads and applies an output projection:

$$\mathrm{MSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \sigma\left(\Big\|_{i=1}^{h} \mathbf{H}_i\right)\mathbf{W}^O, \tag{3}$$

where $\mathbf{H}_i = \mathrm{SA}(\mathbf{Q}\mathbf{W}_Q^i, \mathbf{K}\mathbf{W}_K^i, \mathbf{V}\mathbf{W}_V^i)$, and $\mathbf{W}_Q^i, \mathbf{W}_K^i, \mathbf{W}_V^i$ and $\mathbf{W}^O$ are learnable projection matrices. The resulting representations are subsequently passed through a feed-forward network,

$$\mathrm{FFN}(X) = \phi(XW_1 + b_1)W_2 + b_2,$$

where ϕ is typically ReLU or GELU. Positional information is incorporated through $z_i = X_i + \mathrm{PE}_i$, with $\mathrm{PE}_i \in \mathbb{R}^d$. Each Transformer layer then applies residual connections and layer normalization:

$$\tilde{z}_i = \mathrm{LayerNorm}\big(z_i + \mathrm{MSA}(z_i, z_i, z_i)\big), \quad y_i = \mathrm{LayerNorm}\big(\tilde{z}_i + \mathrm{FFN}(\tilde{z}_i)\big). \tag{4}$$

The attention mechanism above relies on the Euclidean dot product to measure interactions between tokens. In the following subsection, we replace this Euclidean operation with a generalized self-attention mechanism in hyperbolic space and develop its statistical characterization.

3.3 Generalized Attention & Feed-Forward Networks

We now extend the vanilla Transformer to the Poincaré ball $\mathbb{D}_c^d$ by replacing Euclidean operations with their hyperbolic counterparts [19]. This formulation provides a unified geometric framework in which the Euclidean Transformer is recovered as the curvature $c \to 0$ limit. Let $X \in (\mathbb{D}_c^d)^L$

denote a sequence of L token embeddings. Following [62], we define the hyperbolic attention and feed-forward transformations as

$$\mathcal{A}^c(X) = X \oplus_c \exp_0^c \left[\sum_{j=1}^{h} U_f^j U_v^j \log_0^c(X) \sigma(V) \right], \tag{5}$$

$$\mathcal{F}^c(X) = \left[\mathcal{A}^c(X) \oplus_c \exp_0^c(Z) \oplus_c \exp_0^c(l_2 \mathbb{1}_t^T) \right], \tag{6}$$

where

$$V = \left(U_k^j \log_0^c(X) \right)^T U_q^j \log_0^c(X), \quad \text{and} \quad Z = \left[U_2 \mathrm{ReLU} \left(U_1 \log_0^c \left(\mathcal{A}^c(X) \right) + l_1 \mathbb{1}_t^T \right) \right].$$

Here, $U_f^j \in \mathbb{R}^{d \times s}$ concatenates the h heads, while $U_q^j, U_k^j, U_v^j \in \mathbb{R}^{s \times d}$ are the query, key, and value projections. Further, $U_1 \in \mathbb{R}^{r \times d}$, $U_2 \in \mathbb{R}^{d \times r}$ and $l_1 \in \mathbb{R}^r, l_2 \in \mathbb{R}^d$ parameterize the feed-forward transformation. The activation σ denotes softmax. Both transformations employ residual connections through Möbius addition. The corresponding matrix-vector formulation follows from Lemma 3.

Let $t_{\mathcal{H}}^{h,s,r} : (\mathbb{D}_c^d)^L \to (\mathbb{D}_c^d)^L$ denote the resulting hyperbolic Transformer block. Finite compositions of these blocks define the function class

$$\mathcal{T}_{\mathcal{H}}^{h,s,r} := \left\{ f : (\mathbb{D}_c^d)^L \to (\mathbb{D}_c^d)^L \mid f \text{ is a finite composition of } t_{\mathcal{H}}^{h,s,r} \right\}. \tag{7}$$

This class encompasses both encoder and decoder architectures, since each is composed of attention and feed-forward blocks.

Following Theorems 2 and 3 of [61], we incorporate trainable positional encodings to remove the permutation-equivariance restriction. Let $E \in (\mathbb{D}_c^d)^L$ denote hyperbolic positional embeddings. The resulting class is

$$\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r} = \left\{ f_{\mathcal{P}}(X) = f(X \oplus_c E) \mid f \in \mathcal{T}_{\mathcal{H}}^{h,s,r},\ E \in (\mathbb{D}_c^d)^L \right\}. \tag{8}$$

For $(h, s, r) = (2, 1, 4)$, $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$ universally approximates continuous sequence-to-sequence functions on compact domains, and increasing (h, s, r) further enriches the function class. Thus, the proposed hyperbolic architecture retains the universal approximation property while providing a natural extension of the Euclidean Transformer to non-Euclidean representation spaces.

4 Theoretical Analysis

We consider an i.i.d. dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^t$ sampled from an unknown Borel probability measure ρ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y} \subseteq (\mathbb{D}_c^d)^L$ denote input and output sequences of length L embedded in the Poincaré ball. We assume their support is contained in a compact set $\mathcal{R} \subseteq (\mathbb{D}_c^d)^L$. Our goal is to learn a sequence-to-sequence map $f : (\mathbb{D}_c^d)^L \to (\mathbb{D}_c^d)^L$ directly from ρ, without imposing a parametric model on the data-generating process.

We measure prediction error through the hyperbolic squared distance, in either of the following two forms, which are anyway equal according to Lemma 3, henceforth we work with $\mathcal{E}(f) := \mathcal{E}_1(f)$:

$$\mathcal{E}_1(f) := \int_{\mathcal{X} \times \mathcal{Y}} \| \log_0^c(f(x)) - \log_0^c(y) \|^2 \, d\rho(x, y), \quad \text{or} \quad \mathcal{E}_2(f) := \int_{\mathcal{X} \times \mathcal{Y}} d_c(f(x), y)^2 \, d\rho(x, y). \tag{9}$$

The population-optimal predictor is the *Hyperbolic Regression Function (HRF)*, $f_\rho(x) := \exp_0^c \left(\int_{\mathcal{Y}} \log_0^c(y) \, d\rho(y|x) \right)$ [see Lemma 5], where $\rho(\cdot|x)$ is the conditional distribution of Y given $X = x$. Importantly, f_ρ is defined independently of the chosen model class and thus remains the population target under model misspecification.

Thus, without assuming model correctness, the statistical objective is to universally approximate the population-optimal functional f_ρ using the hyperbolic Transformer class. Given $\mathcal{D}$, the corresponding empirical risk minimizer is

$$f_D^{h,s,r} := \arg \min_{f \in \mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}} \mathcal{E}_D(f), \tag{10}$$

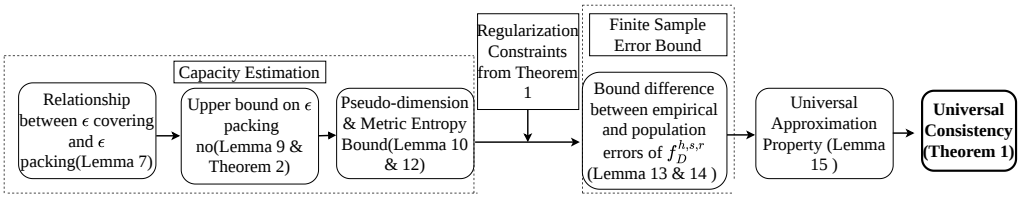


Figure 3: A schematic to understand the flow of theoretical analysis. Lemmas and theorems not mentioned in the main paper can be found in the Supplementary document.

where

$$\mathcal{E}_D(f) := \frac{1}{t}\sum_{i=1}^{\lceil t/L \rceil} \| \log_0^c(f(X_i)) - \log_0^c(Y_i)\|_{L^2}^2. \tag{11}$$

For $c \to 0$, this reduces to the standard Euclidean squared-error risk.

Our central objective is to establish universal statistical (strong) consistency [23], for an appropriately truncated sequence of empirical risk minimizers, i.e., to prove $\mathcal{E}_D\left(\pi_{M_t} f_D^{h,s,r}\right) \longrightarrow \mathcal{E}(f_\rho)$ as $t \to \infty$, and to quantify the corresponding excess-risk rate and its dependence on the input-token embedding dimension d. .

The following theorem offers a statistical guarantee that the optimal estimators $f_D^{h,s,r}$ attain strong consistency in the notion following the definition of [23]. Specifically, it establishes that generalized Transformer architectures, when regarded as sequence-to-sequence regression models and trained under appropriate model constraints, exhibit strong universal consistency.

Theorem 1. *Given $\theta \in (0, 1/2d)$ arbitrary, and number of input samples t, and any fixed input embedding dimension d, if the following conditions hold as $t \to \infty$*

(i) the sequence of the truncation parameters on the input tokens $M_t \to \frac{1}{\sqrt{c}}$, and hence,

(ii) $t^{-\theta} M_t^2 \left[\frac{1}{M_t\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right]^2 \to 0$*, and*

(iii) $\frac{P\log(Q)}{t^{1-2\theta d}} \to 0$*, where*

$$P := \left(\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right)^4 d\log(d),$$

$$Q := dt\left(2\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})t^\theta\right)^{dL} \times \log\left(dt\left(2\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})t^\theta\right)^{dL}\right),$$

then, $\pi_{M_t} f_D^{h,s,r}$ is strongly universally consistent for all triplet $(h, s, r) \geq (2, 1, 4)$, where $\pi_w(u) := \max\{w, |u|\}$ is the well-known truncation operator acting element-wise.

Proof. See Appendix B.

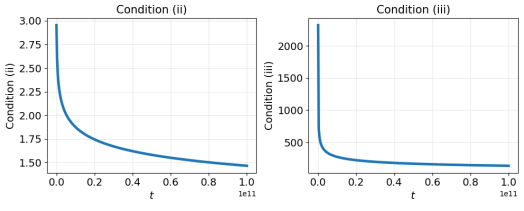


Figure 2: We plot the expressions involved in conditions (ii) and (iii) from Theorem 1, with respect to the number of samples. Note that since the conditions can be analytically shown to be true for any arbitrary choice of sequence $\{M_t\}_{t\in\mathbb{N}}$ with $M_t \to 1/\sqrt{c}$, we have taken a particular choice of $M_t = \frac{1-t^{-1}}{\sqrt{c}}$ to show the empirical convergence.

We pause briefly to examine the analytical and empirical interpretability of the conditions stated above. In particular, we characterize the admissible growth of the truncation sequence M_t and

illustrate how these conditions can be verified analytically as the sequence length t increases. This analysis serves two purposes: it establishes that the stated conditions are analytically feasible and clarifies how the truncation regime affects the complexity and generalization behavior of the model. We discuss both aspects below.

Analytical Feasibility of the conditions in Theorem 1 The conditions in Theorem 1 are non-vacuous and can be satisfied by any suitable choice of the truncation sequence. For example, taking $M_t = \frac{1}{\sqrt{c}}(1 - t^{-\alpha})$ for any $\alpha > 0$ gives $\tanh^{-1}(M_t\sqrt{c}) = O(\log t)$, so that condition (ii) holds for every $\theta > 0$, while condition (iii) satisfies

$$\frac{P \log Q}{t^{1-2\theta d}} = \mathcal{O}\left(\frac{(\log t)^4}{t^{1-2\theta d}}\right) \to 0 \qquad \text{for } \theta < \frac{1}{2d}.$$

Thus, the assumptions are compatible and can be empirically illustrated by evaluating these quantities over increasing sequence lengths t. We refer to Figure 2 for thic specific demonstration and refer to Lemma 6 for a general analytical proof, shoiwng that conditions (ii) & (iii) indeed follow from condition (i).

Interpretation of conditions in Theorem 1 The **first condition**, which involves truncation parameters, describes a specific geometric setup where input tokens are gradually placed closer to the boundary of the Poincaré ball with radius $1/\sqrt{c}$ [more like expanding the vocabulary in the case of a language modelling task]. The **second condition** sets a lower limit on the number of input tokens in relation to how far they are from the center of the ball. This captures a concentration effect: the number of tokens must grow faster than their radial distance, which helps maintain the strong convergence phenomenon accompanied by Borel-Cantelli's criterion. The **third condition** concerns the terms P and Q, representing the pseudo-dimension and the metric entropy of the hypothesis class $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$. This condition ensures that the combined growth of these complexity measures stays small compared to the sequence length, which is necessary for generalization and stable training.

It is worth noting that all the operations involved are continuous with respect to the curvature parameter c [including the empirical risk function defined by Equation 11]. This implies that the sequence of regression estimators $\pi_{M_t} f_D^{h,s,r}$ built in Theorem 1 is also continuous in c. **As a result, the universal consistency of the Euclidean Transformer emerges as a special case of Theorem 1, obtained by taking the limit $\lim_{c\to 0}$, which we defer as Corollary 17 in the Appendix**.

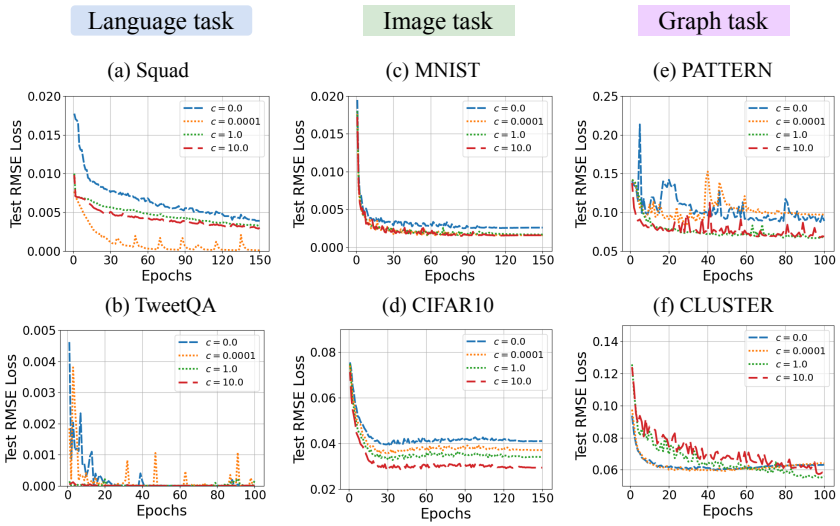
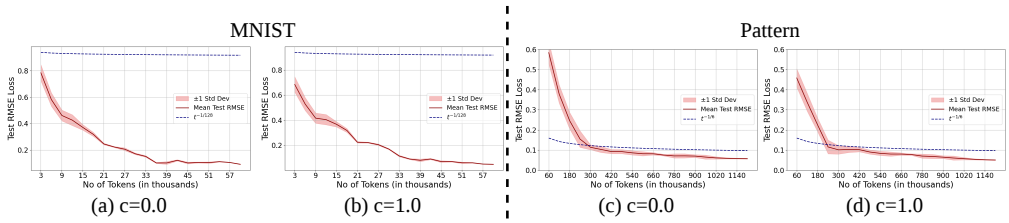


Figure 4: The plots of test RMSE losses for six datasets are presented with various curvature settings. The loss converges faster compared to the Euclidean variant ($c = 0$) at least for one non-zero curvature.

Remark 2. (**Implication of using the truncation parameter M_t in Theorem 1.**) In mathematical analysis, series convergence is typically assessed via partial sums, by examining the sum up to the t-th term and taking the limit as $t \to \infty$. This principle motivates the introduction of the truncation operator in Theorem 1. In contrast to [36], which considers $M_t \to \infty$, we study the regime



(a) c=0.0 (b) c=1.0 (c) c=0.0 (d) c=1.0

Figure 5: Validating theoretical sample complexity bounds $\mathcal{O}(t^{-1/2d})$ [Lemma 16] on MNIST and Pattern datasets for two different curvatures

Table 2: The comparative results for six datasets (two from each category of language, image, and graphs) with varying curvatures are presented. The best results are boldfaced, and the respective metrics are mentioned in parentheses. The model performance became optimal for the high curvature.

Curvatures / Datasets	Squad (F1)	Tweet (Acc)	MNIST (Acc)	CIFAR10 (Acc)	PATTERN (Acc)	CLUSTER (Acc)
$c = 0$ (Euclidean)	0.8439 ± 0.471	$\mathbf{100} \pm 0.00$	99.10 ± 0.020	74.6 ± 1.283	75.0292 ± 0.598	73.5513 ± 0.724
$c = 0.0001$	$\mathbf{0.9943} \pm 0.083$	$\mathbf{100} \pm 0.00$	99.10 ± 0.036	74.08 ± 1.260	74.995 ± 0.482	73.9599 ± 0.801
$c = 1.0$	0.8533 ± 1.429	$\mathbf{100} \pm 0.00$	99.05 ± 0.018	75.15 ± 1.305	81.2558 ± 1.027	76.7312 ± 0.832
$c = 10.0$	0.9067 ± 1.638	$\mathbf{100} \pm 0.00$	$\mathbf{99.13} \pm 0.046$	$\mathbf{75.28} \pm 1.472$	$\mathbf{82.4095} \pm 1.331$	$\mathbf{76.9425} \pm 0.761$

$M_t \to 1/\sqrt{c}$, placing samples near the boundary of the Poincaré Ball $\mathbb{D}_c^d$, whose radius is $1/\sqrt{c}$. As $M_t \to 1/\sqrt{c}$, the term $\tanh^{-1}(M_t\sqrt{c}) \to \infty$, implying that $M_t \left(\frac{1}{M_t\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right) \to \infty$. This asymptotic behavior enables an upper bound on the covering number of $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$ in terms of the truncation threshold. The use of truncation is further justified by its foundational role in establishing the universal consistency of various learning algorithms [23].

We provide a schematic of a proof outline to show the consistency of the Transformers in Figure 3. We first extend the vanilla Transformer to a generalized hyperbolic setup and name it $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$. We deduce an upper bound on the metric entropy of the function class $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$ by extending the results of [26]. In the next stage, taking the cue from the non-parametric Pseudo-dimension bound from Theorem 7 [1], we establish the relation between metric entropy and pseudo-dimension of the class $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$. Thereafter, we derive an upper bound on the difference between the truncated empirical error and the universal (population) error in Lemma 14 by using a generalized version of the concentration inequality from [23]. Finally, combining the results from universal approximation properties [62] and previously derived bounds, we prove the universal consistency of the Transformer class $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$ in Theorem 1. Additionally, we provide a result on sample complexity (which is also derivable from the proof of Theorem 1) in Lemma 16, for clarity and coherence of narrative, we leave a remark in Lemma 16 [with proof in Appendix B.4], while deferring the other technical Lemmas and Theorem 1 with proofs in the Supplementary document.

Lemma 16. (**Sample Complexity**) For any finite truncation parameter M, the empirical error converges to the universal error at a rate of $\mathcal{O}(t^{-1/2d})$ with probability at least $1 - C^* P \log(Q) \exp\left(-\frac{t^{1-2d\theta}}{214\times(3M^4)^d}\right)$ (this probability $\to 1$ as $t \to \infty$ by (iii), Theorem 1), where P and Q are as defined in Theorem 1, and $C^* > 0$ is a constant.

5 Experiments

Details of the Datasets. We executed experiments on three different categories of datasets to justify the theoretical analysis. We considered Squad [47] and TweetQA [60] for language tasks, MNIST [12] and CIFAR10 [32] for image tasks, and PATTERN and CLUSTER [18] for graph tasks.

Experimental Setup. We applied Language Transformer [57] on SQuAD and Tweet, Vision Transformer [15] on MNIST and CIFAR10, and Graph Transformer [16] on PATTERN and

CLUSTER. We strictly adhere to the standard architectures for all three Transformers. The language transformer consists of both encoders and decoders. On the other hand, ViT has only encoders, as per the proponents. Furthermore, GT has the encoder and a series of feedforward blocks as decoders. The Transformers are trained on the RMSE loss, as our entire theoretical analysis encompasses the regression problem. We train each Transformer with four different sets of curvatures, such as $\{0.0, 0.0001, 1.0, 10.0\}$. The details of the hyperparameters are provided in Table 3 of the Supplementary document. Our code base is available at https://anonymous.4open.science/r/transformerConsistency-0347/README.md

Observation & Discussion. We presented test RMSE loss in Figure 4 for six datasets belonging to three distinct categories: language, image, and graph. The patterns observed in the loss curves indicate faster convergence compared to the Euclidean counterparts, where $c = 0$. Precisely, for every dataset, we can find at least one non-zero curvature for which the Transformer attained faster convergence in terms of RMSE loss. Such properties are broadly aligned with the theoretical claims validated across various token categories. In some cases (such as TweetQA with $c = 10.0$), the test loss exhibited an early-stage saturation problem for higher curvatures. The occurrence of such cases is due to the contraction of Poincaré ball with a high curvature.

The results demonstrated in Table 2 indicate a consistent benefit from increasing curvature, particularly for the image and graph datasets. While the Euclidean setting ($c = 0$) performs competitively on MNIST and CIFAR-10, increasing the curvature to $c = 10$ yields the best performance on four of the six datasets, with notable gains on PATTERN and CLUSTER. For language data, the effect is more dataset-dependent, with SQuAD achieving its best performance at a small curvature ($c = 0.0001$), while Tweet maintains perfect accuracy across all settings.

Sample Complexity. We conducted experiments across all six datasets to empirically validate the sample complexity rate $\mathcal{O}(t^{-1/2d})$ established in Lemma 16. For each dataset, the available training tokens were divided into 20 equally sized partitions, and the number of training tokens was progressively increased across these partitions. In each setting, we trained the Transformer models under two curvature regimes, namely $c = 0.0$ and $c = 1.0$. Figure 5 illustrates the resulting behavior for two representative datasets. Notably, the MNIST image dataset has an input embedding dimension of $d = 64$, whereas the PATTERN graph dataset has an input embedding dimension of $d = 3$. Consequently, the theoretical sample complexity bound is substantially tighter for PATTERN than for MNIST, as reflected by the dependence of the rate on the embedding dimension d. For both datasets, we report the test RMSE over 15 independent random seeds. The resulting empirical RMSE curves asymptotically fall below the corresponding theoretical curves, providing empirical support for the sample complexity rate established in Lemma 16. We further refer to Appendix C.2 for the same validation on the remaining four datasets.

6 Conclusion & Future Works

In this work, we developed a unified statistical framework for sequence-to-sequence function regression with generalized Transformers on non-Euclidean domains. By embedding the Transformer architecture within the Poincaré ball, we derived metric entropy and truncation error bounds for the class $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$. Combined with the universal approximation properties of Transformers, these results establish universal consistency and yield explicit sample complexity guarantees.

A key takeaway is that our framework provides a common statistical characterization across language, image, and graph-based Transformers, including their hyperbolic variants, through an appropriate abstraction of tokens. Empirical results across these modalities further support the predicted sample complexity behavior, highlighting the role of representation dimension in the statistical efficiency of Transformer learning.

Moreover, exploring connections between formal language theory and the statistical behavior of Transformer architectures presents a compelling avenue for future research, particularly in non-i.i.d. settings or under adversarial perturbations. Another promising direction lies in establishing dimension-independent sample complexity bounds for the class $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$, as the bounds obtained in Lemma 16 are still subject to the well-known *curse of dimensionality*.

AI USE STATEMENT

In this work, we used generative AI tools for polishing and improving the clarity, grammar, and presentation of the manuscript. We have not used generative AI tools for generating research results, conducting experiments, performing statistical analyses, or making substantive scientific claims. All mathematical results, theoretical arguments, experimental designs, analyses, and conclusions were independently developed and verified by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work does not have any societal concerns and does not violate any ethical norms maintained by ICLR 2027.

REPRODUCIBILITY STATEMENT

We claim that all the experimental simulations provided are reproducible. As evidence, we have provided the anonymous GitHub link to verify it.

REFERENCES

[1] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[2] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

[3] Pablo Barceló, Alexander Kozachinskiy, Anthony Widjaja Lin, and Vladimir Podolskii. Logical languages accepted by transformer encoders with hard attention. *arXiv preprint arXiv:2310.03817*, 2023.

[4] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[5] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[6] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*, 2020.

[7] Sebastian G Bouschery, Vera Blazevic, and Frank T Piller. Augmenting human innovation teams with artificial intelligence: Exploring transformer-based language models. *Journal of Product Innovation Management*, 40(2):139–153, 2023.

[8] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[9] David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *International Conference on Machine Learning*, pp. 5544–5562. PMLR, 2023.

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

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

[12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

[14] M.P. do Carmo. *Riemannian Geometry*. Mathematics (Boston, Mass.). Birkhäuser, 1992. ISBN 9783764334901.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[17] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

[18] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[19] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

[20] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[21] Cheng Gao, Yuan Cao, Zihao Li, Yihan He, Mengdi Wang, Han Liu, Jason Klusowski, and Jianqing Fan. Global convergence in training large-scale transformers. *Advances in Neural Information Processing Systems*, 37:29213–29284, 2024.

[22] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[23] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[24] Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, 7(10):992, 2019.

[25] Yiding Hao, Dana Angluin, and Robert Frank. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810, 2022.

[26] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.

[27] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

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

[29] Ruiquan Huang, Yingbin Liang, and Jing Yang. Non-asymptotic convergence of training transformers for next-token prediction. *arXiv preprint arXiv:2409.17335*, 2024.

[30] Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? *arXiv preprint arXiv:2307.14023*, 2023.

[31] Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404. PMLR, 2020.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

[33] Serge Lang. *Differential and Riemannian manifolds*. Springer Science & Business Media, 1995.

[34] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[35] Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. *Advances in neural information processing systems*, 31, 2018.

[36] Shao-Bo Lin, Kaidong Wang, Yao Wang, and Ding-Xuan Zhou. Universal consistency of deep convolutional neural networks. *IEEE Transactions on Information Theory*, 68(7):4610–4617, 2022. doi: 10.1109/TIT.2022.3151753.

[37] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

[38] William Merrill and Ashish Sabharwal. A logic for expressing log-precision transformers, 2023. URL `https://arxiv.org/abs/2210.02671`.

[39] Taro Miyazaki, Hideya Mino, and Hiroyuki Kaneko. Understanding how positional encodings work in transformer model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 17011–17018, 2024.

[40] Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

[41] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, 2017.

[42] Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023.

[43] Reese Pathak, Rajat Sen, Weihao Kong, and Abhimanyu Das. Transformers can optimally learn regression mixture models. *arXiv preprint arXiv:2311.08362*, 2023.

[44] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

[45] Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021.

[46] Guo Qipeng, Qiu Xipeng, Liu Pengfei, Shao Yunfan, Xue Xiangyang, and Zhang Zheng. Star-transformer. In *Proceedings of HLT-NAACL*, pp. 1315–1325, 2019.

[47] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[48] Akiyoshi Sannai, Yuuki Takai, and Matthieu Cordonnier. Universal approximations of permutation invariant/equivariant functions by deep neural networks. *arXiv preprint arXiv:1903.01939*, 2019.

[49] Taihua Shao, Yupu Guo, Honghui Chen, and Zepeng Hao. Transformer-based neural network for answer selection in question answering. *IEEE Access*, 7:26146–26156, 2019.

[50] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

[51] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pp. 9438–9447. PMLR, 2020.

[52] Gemini Team, R Anil, S Borgeaud, Y Wu, JB Alayrac, J Yu, R Soricut, J Schalkwyk, AM Dai, A Hauth, et al. Gemini: A family of highly capable multimodal models, 2024. *arXiv preprint arXiv:2312.11805*, 10, 2024.

[53] Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in neural information processing systems*, 36:71911–71947, 2023.

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[55] Loring W Tu. *Differential geometry: connections, curvature, and characteristic classes*, volume 275. Springer, 2017.

[56] Abraham Ungar. *A gyrovector space approach to hyperbolic geometry*. Springer Nature, 2022.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

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

[60] Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. Tweetqa: A social media focused question answering dataset. *arXiv preprint arXiv:1907.06292*, 2019.

[61] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

[62] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions?, 2020. URL https://arxiv.org/abs/1912.10077.

[63] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[64] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2):787–794, 2020.

Appendix

CONTENTS

1 Introduction **1**

2 Related Works **2**

3 Extending Framework to a Generalized Set-up **3**

3.1 A Motivating Example 3

3.2 Preliminaries on Vanilla Transformers 4

3.3 Generalized Attention & Feed-Forward Networks 4

4 Theoretical Analysis **5**

5 Experiments **8**

6 Conclusion & Future Works **9**

A Fundamentals of Riemannian Manifolds and Hyperbolic Geometry **14**

B Detailed Proof of Theorem 1 **17**

B.1 Step 1: Capacity Estimates for a Class of Functions Characterized by $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$: . . . 18

B.2 Step 2: Finite Sample Error Bound: 22

B.3 Step 3: Universal Approximations of Transformers: 23

B.4 Other Technical proofs 24

C Additional Experiments and Details **26**

C.1 Additional Results 26

C.2 Additional Experiments on Sample Complexity 26

C.3 Hyperparameter Details 26

C.4 Computational Resources 27

A FUNDAMENTALS OF RIEMANNIAN MANIFOLDS AND HYPERBOLIC GEOMETRY

Riemannian Manifold, Tangent Space, and Geodesics. An *n-dimensional manifold* $\mathcal{M}$ is a topological space locally homeomorphic to $\mathbb{R}^n$ [55]. At each $x \in \mathcal{M}$, the *tangent space* $T_x(\mathcal{M})$ linearly approximates $\mathcal{M}$. A *Riemannian manifold* is a manifold with a collection of smoothly varying metric tensors $g_x : T_x(\mathcal{M}) \times T_x(\mathcal{M}) \to \mathbb{R}$ [14], inducing a distance between points $p, q \in \mathcal{M}$ via a piecewise smooth curve $\gamma : [a, b] \to \mathcal{M}$:

$$L(\gamma) := \int_a^b \sqrt{g_{\gamma(t)}(\gamma'(t), \gamma'(t))}\, dt.$$

A *geodesic* minimizes $L(\gamma)$ between p and q defines the *geodesic distance* between them.

Given linearly independent $u, v \in T_x(\mathcal{M})$, the *sectional curvature* at x is

$$k_x(u,v) := \frac{g_x(R(u,v)v,u)}{g_x(u,u)g_x(v,v) - g_x(u,v)^2},$$

where R is the curvature tensor: $R(u,v)w = \nabla_u\nabla_v w - \nabla_v\nabla_u w - \nabla_{[\nabla_u v - \nabla_v u]}w$, with ∇ the Riemannian connection.

An n-dimensional *hyperbolic space* is a connected, complete Riemannian manifold with constant negative sectional curvature. Models like the Poincaré Ball, Poincaré half-space, Klein-Beltrami, and hyperboloid are all isometric by the Killing–Hopf Theorem [33]. We use the Poincaré Ball model and outline its core operations below.

Poincaré Ball Model. With curvature $-c$ ($c > 0$), the Poincaré Ball $\mathbb{D}_c^n$ consists of points in the ball of radius $1/\sqrt{c}$ in $\mathbb{R}^n$ [34]. Geodesics are circular arcs orthogonal to the boundary. The induced distance by the Poincaré conformal factor between $u, v \in \mathbb{D}_c^n$ is:

$$d_c(u,v) \coloneqq \frac{1}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\| - u \oplus_c v\|\right), \tag{12}$$

where $\oplus_c$ is defined as the M*ö*bius Addition [see Equation 13].

Gyrovector Space. Gyrovector spaces [56] extend vector space concepts to hyperbolic geometry using gyrogroup theory. Möbius addition and scalar multiplication on $\mathbb{D}_c^n$ are key operations:

Möbius Addition For $u, v \in \mathbb{D}_c^n$,

$$u \oplus_c v := \frac{(1 + 2c\langle u, v\rangle + c\|v\|^2)u + (1 - c\|u\|^2)v}{1 + 2c\langle u, v\rangle + c^2\|u\|^2\|v\|^2}. \tag{13}$$

Möbius Scalar Multiplication For $r \in \mathbb{R}$, $u \in \mathbb{D}_c^n$,

$$r \otimes_c u := \frac{1}{\sqrt{c}} \tanh\left(r \tanh^{-1}(\sqrt{c}\|u\|)\right) \frac{u}{\|u\|}. \tag{14}$$

The induced distance (12), addition (13), and multiplication (14) defined above get reduced to the analogous notions of their respective Euclidean versions, once we note that

$$\lim_{c\to 0} d_c(u,v) = \| - u + v\|, \lim_{c\to 0} u \oplus_c v = u + v, \text{ and}$$
$$\lim_{c\to 0} r \otimes_c u = ru.$$

Exponential & Logarithmic Maps. For $x \in \mathbb{D}_c^n$, the exponential map $\exp_x^c : T_x(\mathbb{D}_c^n) \to \mathbb{D}_c^n$ and logarithmic map $\log_x^c : \mathbb{D}_c^n \to T_x(\mathbb{D}_c^n)$ are:

$$\exp_x^c(v) := x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right), \tag{15}$$

and

$$\log_x^c(y) := \frac{2}{\sqrt{c}\lambda_x^c} \tanh^{-1}\left(\sqrt{c}\|-x \oplus_c y\|\right) \frac{-x \oplus_c y}{\|-x \oplus_c y\|}, \tag{16}$$

with conformal factor $\lambda_x^c := \frac{2}{1-c\|x\|^2}$. The exponential map projects a point on the tangent space (in $\mathbb{R}^n$) to a point on the Poincaré Ball, while the logarithm map (as an inverse operation) projects a point from the Poincareé Ball to the tangent space.

M*ö*bius Matrix Vector Multiplication. For any $x \in \mathbb{D}_c^n$ and $A \in \mathbb{R}^{n\times m}$, the M*ö*bius matrix vector multiplication is defined as:

$$A^{\otimes_c} x \coloneqq \frac{1}{\sqrt{c}} \tanh\left(\frac{\|Ax\|}{\|x\|} \tanh^{-1}(\sqrt{c}\|x\|)\right) \frac{Ax}{\|Ax\|}. \tag{17}$$

Lemma 3. *For a linear transformation $A : \mathbb{R}^m \to \mathbb{R}^n$ [where we identify A with its matrix representation], and for any $x \in \mathbb{D}_c^m$, $A^{\otimes_c} x = \exp_0^c(A \log_0^c(x))$.*

Proof. From Equation 16, we write

$$\log_0^c(x) = \frac{1}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|x\|)\frac{x}{\|x\|}.$$

Hence,

$$A\log_0^c(x) = \frac{1}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|x\|)\frac{Ax}{\|x\|}.$$

Finally, using Equation 15, we get

$$\begin{aligned}\exp_0^c(A\log_0^c(x)) &= \frac{1}{\sqrt{c}} \tanh\left(\frac{Ax}{\|x\|} \tanh^{-1}(\sqrt{c}\|x\|)\right)\frac{Ax}{\|Ax\|} \\ &= A^{\otimes_c}x.\end{aligned}$$

□

Lemma 4. *For any two points in $p, q \in \mathbb{D}_c^n$ (where n can be any positive integer),*

$$d_c(p,q) = \|\log_0^c(p) - \log_0^c(q)\| \tag{18}$$

Proof. Consider the geodesics $\mathcal{G}_1$ connecting 0 and p and $\mathcal{G}_2$ connecting 0 and q on $\mathbb{D}_c^n$. Then, $\log_0^c(p)$ and $\log_0^c(q)$ are the respective projections of $\mathcal{G}_1$ and $\mathcal{G}_2$ on $T_0(\mathbb{D}_c^n)$. Since $T_0(\mathbb{D}_c^n) \cong \mathbb{R}^n$, $\log_0^c(q) - \log_0^c(p) \in T_0(\mathbb{D}_c^n)$. Now the parallel transport of $\log_0^c(q) - \log_0^c(p)$ w.r.t. the Riemannian Connection of $\log_0^c(q) - \log_0^c(p)$ to $T_p(\mathbb{D}_c^n)$ is given by [see Theorem 4, [20]]

$$\begin{aligned}&\text{P.T.}_{0\to p}^c\left(\log_0^c(q) - \log_0^c(p)\right) \\ =&\log_0^c\left(p \oplus_c \exp_0^c\left(\log_0^c(q) - \log_p^c(p)\right)\right) \\ =&\log_p^c(p \oplus_c (q \oplus_c (-p))) \\ =&\log_p^c(q)\end{aligned} \tag{19}$$

Since the parallel transport is an isometrically invariant operation,

$$\|\log_p^c(q)\| = \|\log_0^c(q) - \log_0^c(p)\|. \tag{20}$$

Similarly, a parallel transport of $\log_p^c(q)$ to $T_0(\mathbb{D}_c^n)$ w.r.t. the Riemannian Connection of $\log_p^c(q)$ gives us

$$\begin{aligned}&\text{P.T.}_{p\to 0}^c\left(\log_p^c(q)\right) \\ =&\frac{\lambda_p^c}{\lambda_0^c}\log_p^c(q) \\ =&\frac{1}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\|-p \oplus_c q\|\right)\frac{-p \oplus_c q}{\|-p \oplus_c q\|}.\end{aligned} \tag{21}$$

But according to Equation (12),

$$\begin{aligned}d_c(p,q) &= \left\|\frac{1}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\|-p \oplus_c q\|\right)\frac{-p \oplus_c q}{\|-p \oplus_c q\|}\right\| \\ &= \|\log_p^c(q)\| \\ &= \|\log_0^c(q) - \log_0^c(p)\|,\end{aligned}$$

where the last two equalities are followed by the isometry of the parallel transport and from Equation 21, completing the proof of Lemma 2. □

Lemma 5. *There exists an optimal solution to (9), given by the Hyperbolic Regression Function (HRF) $f_\rho(x) := \exp_0^c\left(\int_{\mathcal{Y}} \log_0^c(y) d\rho(y|x)\right)$, where $\rho(\cdot|x)$ is the conditional distribution corresponding to ρ given $x \in \mathcal{X}$.*

Proof. Given the L^2 estimation error

$$\begin{aligned}\mathcal{E}(f) &= \int_{\mathcal{Z}} \|\log_0^c(f(x)) - \log_0^c(y)\|^2 d\rho \\ &= \mathbb{E}_{\mathcal{X},\mathcal{Y}}[\|\log_0^c(f(\mathcal{X})) - \log_0^c(\mathcal{Y})\|]^2,\end{aligned}$$

for any function $g : \mathcal{X} \to \mathbb{R}^d$, $\mathcal{E}(g)$ can be written in terms of conditional expectation as

$$\begin{aligned}\mathcal{E}(g) =& \mathbb{E}_{\mathcal{X},\mathcal{Y}}\left[\|g(\mathcal{X}) - \log_0^c(\mathcal{Y})\|^2\right] \\ &= \mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\|g(\mathcal{X}) - \mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X}\|^2\right] + \mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\|\mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X} - \log_0^c(\mathcal{Y})\|^2\right] \\ &+ 2\mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\left[g(\mathcal{X}) - \mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X}\right]^t \times \left[\mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X} - \log_0^c(\mathcal{Y})\right]\right]\end{aligned}$$

The cross term in the last expression is 0, since $\mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\left[\mathbb{E}(\log_0^c(\mathcal{Y})) - \log_0^c(\mathcal{Y})\right]\right] = 0$. Therefore, the expression for $\mathcal{E}(g)$ is reduced to

$$\mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\|g(\mathcal{X}) - \mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X}\|^2\right] + \mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\|\mathbb{E}(\log_0^c(\mathcal{Y}))|\mathcal{X} - \log_0^c(\mathcal{Y})\|^2\right]$$

which attains minimum at $g(x) = \mathbb{E}\left[\log_0^c(\mathcal{Y})|x\right]$ for each $x \in \mathcal{X}$. Hence, we write for each $x \in \mathcal{X}$, $g(x) = \int_{\mathcal{Y}} \log_0^c(y) d\rho(y|x)$. Following our conventions, $g(x) = \log_0^c(f(x))$, and hence, $f(x) = \exp_0^c\left(\int_{\mathcal{Y}} \log_0^c(y) d\rho(y|x)\right)$ □

B Detailed Proof of Theorem 1

Lemma 6. *[Feasibility of Conditions in Theorem 1] Let $\theta \in (0, 1/(2d))$ and let $\{M_t\}_{t\geq 1}$ satisfy*

$$0 < M_t < \frac{1}{\sqrt{c}}, \qquad M_t \longrightarrow \frac{1}{\sqrt{c}}.$$

Define

$$\delta_t := 1 - M_t\sqrt{c} \qquad \textit{and} \qquad L_t := \tanh^{-1}(M_t\sqrt{c}).$$

Then $\delta_t \downarrow 0$ and

$$L_t = \frac{1}{2}\log\frac{2-\delta_t}{\delta_t} = \frac{1}{2}\log\frac{1}{\delta_t} + O(1).$$

Consequently, condition (ii) *is equivalent, up to multiplicative constants, to*

$$t^{-\theta}\log^2\left(\frac{1}{\delta_t}\right) \longrightarrow 0.$$

Moreover, if $t^{-\theta}\log^2\left(\frac{1}{\delta_t}\right) \longrightarrow 0, \quad \theta < \frac{1}{2d}$, then condition (iii) *is also satisfied.*

Proof. Since $M_t\sqrt{c} \to 1$, we can write $M_t\sqrt{c} = 1-\delta_t, \quad \delta_t \to 0^+$. Using $\tanh^{-1}(x) = \frac{1}{2}\log\frac{1+x}{1-x}$, we obtain

$$L_t = \frac{1}{2}\log\frac{2-\delta_t}{\delta_t} = \frac{1}{2}\log\frac{1}{\delta_t} + O(1).$$

Furthermore,

$$M_t^2 = \frac{(1-\delta_t)^2}{c} \asymp \frac{1}{c}, \qquad \frac{L_t}{M_t\sqrt{c}} = \frac{L_t}{1-\delta_t} \asymp L_t.$$

Hence the left-hand side of condition (ii) satisfies

$$t^{-\theta}M_t^2\left[\frac{L_t}{M_t\sqrt{c}}\right]^2 \asymp t^{-\theta}L_t^2 \asymp t^{-\theta}\log^2\left(\frac{1}{\delta_t}\right).$$

Thus condition (ii) holds precisely when

$$t^{-\theta}\log^2\left(\frac{1}{\delta_t}\right) \to 0,$$

which is always true since $\theta \in (0, 1/2d)$.

We next verify the condition (iii). Since $P = \left(\frac{L_t}{\sqrt{c}}\right)^4 d \log d$, we have $P = O(L_t^4)$. Also, defining $A_t := 2\frac{L_t}{\sqrt{c}}\, t^\theta$, the quantity Q satisfies

$$\log Q = \log(dt) + d \log A_t + \log(\log(dt) + d \log A_t)\,.$$

Since $L_t \geq 1$ eventually and $\theta > 0$,

$$\log A_t = \theta \log t + \log L_t + O(1),$$

and therefore

$$\log Q = O(\log t + \log L_t)\,.$$

Under condition (ii), $L_t = o(t^{\theta/2})$, so in particular $\log L_t = O(\log t)$, and hence $\log Q = O(\log t)$. It follows that

$$\frac{P \log Q}{t^{1-2\theta d}} = O\left(\frac{L_t^4 \log t}{t^{1-2\theta d}}\right).$$

Thus condition (iii) is guaranteed whenever

$$L_t^4 \log t = o\big(t^{1-2\theta d}\big)\,,$$

or equivalently,

$$\log^4\left(\frac{1}{\delta_t}\right) \log t = o\big(t^{1-2\theta d}\big)\,,$$

which is again true by the choice of θ. Therefore, no particular parametric choice of M_t is required: any sequence approaching $1/\sqrt{c}$ whose boundary gap δ_t satisfies the above asymptotic growth conditions fulfills the assumptions of Theorem 1. □

B.1 Step 1: Capacity Estimates for a Class of Functions Characterized by $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$:

We begin by adapting certain definitions and terminologies to align with the hyperbolic setting. To facilitate a covering argument for bounding the metric entropy of the Transformer class, we first review key foundational concepts.

Let $\rho \in \mathscr{P}(\mathcal{X})$ and let $f : \mathcal{X} \to \mathbb{D}_c^K$ be a measurable function. The L^p-norm is defined as

$$\|f\|_{L^p(\rho)} := \left[\int_{\mathcal{X}} [\log_0^c(f(x))]^p \, d\rho\right]^{1/p},$$

where for a vector $v = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^n$ and $p \geq 1$, the notation v^p refers to $\sum_{i=1}^n |v_i|^p$. We denote by $L^p(\rho)$ the space of functions $f : \mathcal{X} \to \mathbb{D}_c^K$ such that $\|f\|_{L^p(\rho)} < \infty$. Given a function class $T \subseteq L^p(\rho)$, we define $\mathcal{N}(\epsilon, T, \|\cdot\|_{L^p(\rho)})$ as the ϵ-covering number with respect to the $L^p(\rho)$ norm—i.e., the minimal number of ϵ-radius balls required to cover T. For a sample $v_1^m := v_1, v_2, \ldots, v_m \in \mathcal{X}^m$, let ρ_m denote the empirical distribution over the sample. We then define the empirical covering number as $\mathcal{N}_p(\epsilon, T, v_1^m) := \mathcal{N}(\epsilon, T, \|\cdot\|_{L^p(\rho_m)})$. Similarly, we define the ϵ-packing number $\mathcal{M}(\epsilon, T, \|\cdot\|_{L^p(\rho)})$ as the largest cardinality of a subset $f_1, \ldots, f_{n_p} \subset T$ such that $\|\log_0^c(f_j) - \log_0^c(f_k)\|_{L^p(\rho)} \geq \epsilon$ for all $1 \leq j < k \leq n_p$. The empirical version is denoted $\mathcal{M}_p(\epsilon, T, v_1^m) := \mathcal{M}(\epsilon, T, \|\cdot\|_{L^p(\rho_m)})$.

The following lemma restates the classical relationship between covering and packing numbers.

Lemma 7 ([23]). *For a class of functions T on $\mathcal{X}$, equipped with a probability measure ρ, we have for $p \geq 1$ and $\epsilon > 0$; $\mathcal{M}(2\epsilon, T, \|\cdot\|_{L^p(\rho)}) \leq \mathcal{N}(\epsilon, T, \|\cdot\|_{L^p(\rho)}) \leq \mathcal{M}(\epsilon, T, \|\cdot\|_{L^p(\rho)})$. Given $v_1^m \in \mathcal{X}^m$, we have in particular $\mathcal{M}_p(2\epsilon, T, v_1^m) \leq \mathcal{N}_p(\epsilon, T, v_1^m) \leq \mathcal{M}_p(\epsilon, T, v_1^m)$.*

We now extend the notion of Pseudo-Dimension (Pdim), traditionally defined for real-valued function classes, to functions mapping into the Poincaré Ball. Establishing such a complexity measure is essential for analyzing the expressive capacity of the Transformer class in hyperbolic space.

Definition 8 (Pseudo-Dimension)**.** For $U \subseteq L^p(\rho)$, where $\rho \in \mathscr{P}(\mathcal{X})$, by the Pseudo-Dimension of U, we indicate the largest integer l (denoted as $\text{Pdim}(U)$) for which there exists $(x_1, x_2, ..., x_l, \eta_1^1, \eta_2^1, ..., \eta_1^1, ..., \eta_1^K, \eta_2^K, ..., \eta_l^K) \in \mathcal{X}^l \times \mathbb{R}^{Kl}$ such that for any $(a_1, a_2, ..., a_l) \in \{0,1\}^l$, there exists some $f \in U$ so that for all $i \in \{1, 2, ..., l\}$, $\log_0^c(f(x_i)) > \eta_i = [\eta_1^i, \eta_2^i, ..., \eta_l^i]^T \iff a_i = 1$.

Previously, [36], building on the results of [26], established an upper bound on the ϵ-packing number for real-valued function classes. Here, we extend this framework through Lemma 5 and Theorem 2, leveraging the notion of Pseudo-Dimension. This generalization is particularly advantageous, as it circumvents the curse of dimensionality by remaining insensitive to the ambient (Hausdorff) dimension.

Lemma 9. *Let F be a family of functions on $\mathcal{X}$ (equipped with a probability measure ρ) to $\mathbb{D}_c^K$ satisfying $\log_0^c(f) \in [-M, M]^K$, for all $f \in F$, where $M \in (0, 1/\sqrt{c})$. Let $r := [r_{ij}]_{m \times K}$ be a random matrix where r_{ij} follows Unif$[-M, M]$ i.i.d. Also, let $v := \{v_1, ..., v_m\}$ be a random vector in $\mathcal{X}^m$, where each v_i is drawn independently at random following ρ. Then for all $\epsilon > 0$,*

$$\mathbb{E}\left(|sign((\log_0^c(F))|_v - r)|\right) \geq \mathcal{M}\left(\epsilon, F, \|\cdot\|_{L^p(\rho)}\right) \times \left(1 - \mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)}) e^{-m\left(\frac{\epsilon}{2M}\right)^K}\right), \text{ where}$$

$$\log_0^c(F)|_v \coloneqq \{\log_0^c(f)_v : f \in F\} = \{(\log_0^c(f(v_1)), \log_0^c(f(v_2)), ..., \log_0^c(f(v_m))) : f \in F\}.$$

Proof of Lemma 9. We take G to be an ϵ-separated subset of F with $|G| = \mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)})$. Now,

$$\begin{aligned}
&\mathbb{E}\left(|\text{sign}((\log_0^c(F))|_v - r)|\right) \\
&\geq \mathbb{E}\left(|\text{sign}((\log_0^c(G))|_v - r)|\right) \\
&\geq \mathbb{E}(|\{f \in G : \text{sign}(\log_0^c(f)|_v - r) \neq \text{sign}(\log_0^c(g)|_v - r) \text{ for all } g \in G, g \neq f\}|)) \\
&= \sum_{f \in G} \mathbb{P}(\text{sign}(\log_0^c(f)|_v - r) \neq \text{sign}(\log_0^c(g)|_v - r) \text{ for all } g \in G, g \neq f) \\
&= \sum_{f \in G} (1 - \mathbb{P}(\exists g \in G, g \neq f : \text{sign}(\log_0^c(f)|_v - r) = \text{sign}(\log_0^c(g)|_v - r))) \\
&\geq \sum_{f \in G} (1 - |G| \max_{g \in G, g \neq f} \mathbb{P}(\text{sign}(\log_0^c(f)|_v - r) = \text{sign}(\log_0^c(g)|_v - r))). \qquad (22)
\end{aligned}$$

If f and g are ϵ-separated in G; both $\log_0^c(f), \log_0^c(g) \in [-M, M]^K$, and r_i is drawn at random from the uniform distribution on $[-M, M]^K$, then the probability that $\text{sign}(\log_0^c(f)|_{v_i} - r_i) \neq \text{sign}(\log_0^c(g)|_{v_i} - r_i)$ will be at least $(\epsilon/2M)^K$, for $1 \leq i \leq m$. Hence, we get

$$\mathbb{P}(\text{sign}(\log_0^c(f)|_v - r) = \text{sign}(\log_0^c(g)|_v - r)) \leq \left(1 - \left(\frac{\epsilon}{2M}\right)^K\right) \leq e^{-m\left(\frac{\epsilon}{2M}\right)^K}. \qquad (23)$$

Putting the bounds in Equation (22) and Equation (23) together and by noting that $|G| = \mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)})$, we get

$$\mathbb{E}\left(|\text{sign}((\log_0^c(F))|_v - r)|\right) \geq \mathcal{M}\left(\epsilon, F, \|\cdot\|_{L^p(\rho)}\right) \times \left(1 - \mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)}) e^{-m\left(\frac{\epsilon}{2M}\right)^K}\right).$$

This concludes the proof. □

We now use Lemma 5 to derive an upper bound on the ϵ-covering number of F.

Theorem 2. For a family of functions F with $\text{Pdim}(F) = d$, we have for any $0 < \epsilon \leq 2M$

$$\mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)}) < 2\left[2eK\left(\frac{2M}{\epsilon}\right)^K \log\left(2eK\left(\frac{2M}{\epsilon}\right)^K\right)\right]^d.$$

Proof of Theorem 2. By Sauer's Lemma, we can write

$$|\text{sign}(\log_0^c(F)|_v - r)| \leq \left(\frac{emK}{d}\right)^d,$$

for all $m \geq d$ and $r \in [-M, M]^{mK}$. Given $|G| = \mathcal{M}(\epsilon, F, \|\cdot\|_{L^p(\rho)})$, we get for all $m \geq d$,

$$\left(\frac{emK}{d}\right)^d \geq |G| \left(1 - |G| e^{-m\left(\frac{\epsilon}{2M}\right)^K}\right). \tag{24}$$

It is easy to verify the claim of Theorem 2 if $\left(2M\epsilon^{-1}\right)^K \log(2|G|) < d$ simply by noting that $\epsilon \leq 2M$. So, to prove our claim, we can assume that $\left(2M\epsilon^{-1}\right)^K \log(2|G|) \geq d$. Hence, $m \geq \left(2M\epsilon^{-1}\right)^K \log(2|G|) \geq d$. Simplifying the term on the right in inequality (24), we get

$$\left(1 - |G| e^{-m\left(\frac{\epsilon}{2M}\right)^K}\right) \geq \frac{1}{2}.$$

Hence, using Sauer's Lemma, we have

$$\left(\frac{emK}{d}\right)^d \geq \frac{1}{2}|G|. \tag{25}$$

This is true for all $m \geq \left(\frac{2M}{\epsilon}\right)^K \log(2|G|) \geq d$. In particular, this is true for $m = \left(\frac{2M}{\epsilon}\right)^K \log(2|G|)$. Replacing m by $\left(\frac{2M}{\epsilon}\right)^K \log(2|G|)$ in the inequality (25), we get

$$|G| < 2\left(2eK\left(\frac{2M}{\epsilon}\right)^K \log\left(2eK\left(\frac{2M}{\epsilon}\right)^K\right)\right)^d,$$

completing the proof of Theorem 2. □

Theorem 2 provides an upper bound on the ϵ-packing number for the function class F. However, our primary goal is to derive a bound on the logarithm of the covering number, i.e. the metric entropy. To this end, we estimate the Pseudo-Dimension of $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$ in terms of the model's number of neurons and trainable parameters. Combining this with the covering number bound yields the final capacity estimate for the class of functions realizable by $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$.

We now compute the number of trainable parameters in a single functional block of $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$, comprising a learnable $d \times L$ parameters for postional encoding, an attention layer $\mathcal{A}^c$ with parameters h and s, and a feedforward layer $\mathcal{F}^c$ with parameter r, as defined in Equations (7) and (8). Each attention head includes query, key, value, and output projection matrices, each with sd parameters, yielding $4sd$ per head and $4sdh$ in total. The feedforward layer contributes an additional $2dr$ parameters. An input embedding matrix adds dv parameters, where v is the vocabulary size. Thus, for l blocks, the total number of trainable parameters is $n_{param} = l(4sdh + 2dr + dv) + dL + \mathcal{O}(d)$.

As for the neuron count: each attention head uses d input and s output neurons for the query, key, and value computations, plus d neurons for output projection, totaling $4(d + sh)$. The feedforward layer comprises d input, r hidden, and d output neurons, summing to $2d + r$. Finally, the trainable positional encoding requires another d many neurons. Therefore, the total number of neurons across l blocks is $n_{neuron} = l\,(6d + sh + r) + d$.

Remark 3. Adopting the architectural configuration from [61], and for any finite triplet $(h, s, r) \geq (2, 1, 4)$, the total number of trainable parameters n_{param} and neurons n_{neuron} simplify to $C_1 dt$ and $C_2 d$ respectively, where $C_1, C_2 > 0$ are some constants. Since the Transformer class $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$ consists of finitely many compositions of $\mathcal{A}^c(\cdot)$ and $\mathcal{F}^c(\cdot)$ blocks, the total number of trainable parameters and the number of neurons simplify to again $c_1 dt$ and $c_2 d$ respectively, for some constants $c_1, c_2 > 0$.

We now derive an upper bound on the pseudo-dimension of a truncated subclass of $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$, which leads to Lemma 10. Before presenting the lemma, we clarify the VC-dimension result used from [4]. Although the work primarily focuses on piecewise linear neural networks, Theorem 7 applies to general neural network architectures whose activation functions are piecewise polynomial. Since ReLU activations are piecewise polynomial, this theorem is directly applicable to our setting and will be used to obtain the bound below.

Lemma 10. *An upper bound for the Pseudo-Dimension of $\pi_M(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}})$ can be given as*

$$\textit{Pdim}\left(\pi_M(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}})\right) \leq n_{param}\log_2\left(n_{neuron}\right), \tag{26}$$

where $\pi_M(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}) := \{f \in \mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}} : \log_0^c(f) \in [-M, M]^{d\times t}\}$.

Proof of Lemma 10. By combining Theorem 7 from [4] and Theorem 14.1 from [1] we get that,

$$\text{Pdim}\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right) \leq Cn_{param}\log_2\left(n_{neuron}\right),$$

for some constant $C > 0$. But $\pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right) \subseteq \mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$. Hence

$$\text{Pdim}\left(\pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right)\right) \leq \text{Pdim}\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right),$$

completing the proof of Lemma 10. □

We next establish an upper bound on the covering number of $\pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right)$, thereby enabling capacity estimation of functions in this class with respect to the L^2 norm.

Lemma 11. *There exists an absolute constant C such that for any $0 \leq \epsilon \leq M$,*

$$\log \sup_{X_1^t\in\mathcal{X}^t} \mathcal{N}\left(\epsilon, \pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right), X_1^t\right) \leq Cn_{param}\log n_{neuron} \times \log\left[dt\left(\frac{2M}{\epsilon}\right)^d \log\left(dt\left(\frac{2M}{\epsilon}\right)^d\right)\right],$$

where $X_1^t := \{X_1, X_2, ..., X_t\}$ are the input tokens.

Proof of Lemma 11. For a sequence of t tokens $X_1^t \in \mathbb{D}_c^{d\times t}$, we can write from Lemma 4,

$$\mathcal{N}\left(\epsilon, \pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right), X_1^t\right) \leq \mathcal{M}\left(\epsilon, \pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right), X_1^t\right). \tag{27}$$

By writing $K = dL$, Theorem 2 enables us to write

$$\begin{aligned}&\log\left(\mathcal{M}\left(\epsilon, \pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right), X_1^m\right)\right)\\ &\leq C_1 d\text{Pdim}\left(\pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right)\right) \times \log\left[dt\left(\frac{2M}{\epsilon}\right)^{dL}\log\left(dt\left(\frac{2M}{\epsilon}\right)^{dL}\right)\right].\end{aligned} \tag{28}$$

However, from Lemma 6, we have,

$$\text{Pdim}\left(\pi_M(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}})\right) \leq Cn_{param}\log(n_{neuron}). \tag{29}$$

As such, combining inequalities (27), (28), and (29), we get the result of Lemma 7, completing the proof. □

Corollary 12. *In what follows, combining Lemma 11 , and Remark 3, we get the capacity estimate of $\pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right)$ for $(h, s, r) \geq (2, 1, 4)$ as,*

$$\log \sup_{X_1^t\in\mathcal{X}^t} \mathcal{N}\left(\epsilon, \pi_M\left(\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}\right), X_1^t\right) \leq td^2\log(d)\log\left[dt\left(\frac{2M}{\epsilon}\right)^{dL}\log\left(dt\left(\frac{2M}{\epsilon}\right)^{dL}\right)\right].$$

Remark 4. Corollary 12 provides an upper bound on the metric entropy of the covering number for a truncated subset of $\mathcal{T}^{h,s,r}_{\mathcal{P},\mathcal{H}}$, thereby bounding the capacity of the corresponding bounded function class. This result is subsequently employed in Lemma 13 to derive the almost sure convergence of the empirical error to the population error.

B.2 STEP 2: FINITE SAMPLE ERROR BOUND:

We now seek to demonstrate that the empirical error sequence over finite, bounded samples converges to the universal error arising from empirical risk minimization as formulated in Equation (15). To this end, we establish an upper bound on the discrepancy between the truncated empirical error and its universal counterpart provided by Lemma 9, further facilitated by a version of the concentration inequality stated in Lemma 8. Specifically, for any $f \in \mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$, we define $\mathcal{E}_{\pi_M}(f) := \int_{\mathcal{X}\times\mathcal{Y}} \| \log_0^c(f(x))_M - \log_0^c(y_M)\|^2 d\rho$ and $\mathcal{E}_{\pi_M,D}(f) := \frac{1}{t}\sum_{i=1}^t \| \log_0^c(f(X_i))_M - \log_0^c(Y_{i,M})\|^2$, where $\pi_M(z) := z_M := \min\{M, |z|\} \cdot \text{sign}(z)$, for any $z \in \mathbb{R}$ and for z in higher-dimensional spaces, the truncation is done component-wise on vector-valued inputs.

Lemma 13 ([23]). *We consider $\|y\| \leq B$ and $B \geq \frac{1}{\sqrt{c}}$. For a class of functions $f \in U$ from $\mathbb{D}_c^{d\times t} \to \mathbb{D}_c^{d\times t}$ satisfying $\| \log_0^c(f(x))\| \leq B$, we have*

$$\begin{aligned} &Pr\Big[\exists f \in U : \mathcal{E}(f) - \mathcal{E}(f_\rho) - (\mathcal{E}_D(f) - \mathcal{E}_D(f_\rho)) \geq \epsilon_f\Big] \\ &\leq 14 \sup_{X_1^t \in \mathcal{X}^t} \mathcal{N}_1\left(\frac{\beta\epsilon}{20B}, U, X_1^t\right) \exp\left(-\frac{\epsilon^2(1-\epsilon)\alpha t}{214(1+\epsilon)B^4}\right), \end{aligned} \tag{30}$$

for all $t \geq 1$ and $\alpha, \beta > 0$ and $\epsilon \in (0, 1/2d)$, where $\epsilon_f = (\alpha + \beta + \mathcal{E}(f) - \mathcal{E}(f_\rho))$.

Finally, we use Lemma 11 to derive the finite sample error bound in Lemma 14.

Lemma 14. *Given $t^{-\theta} M_t^2 \left[\frac{1}{M_t\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right]^2 \to 0$, and condition (iii) in Theorem 1 holds for $\theta \in (0, 1/2d)$, then for all $(h, s, r) \geq (2, 1, 4)$,*

$$\lim_{t\to\infty} \mathcal{E}_{\pi_{M_t}}\left(\pi_{M_t} f_D^{h,s,r}\right) \xrightarrow{a.s.} \mathcal{E}_{\pi_{M_t},D}\left(\pi_{M_t} f_D^{h,s,r}\right).$$

Proof of Lemma 14. We know that, $\log_0^c(f_D^{h,s,r}) \in \left[-\frac{1}{\sqrt{c}} \tanh^{-1}(\sqrt{c}), \frac{1}{\sqrt{c}} \tanh^{-1}(\sqrt{c})\right]^d$. Hence, $\left\|\pi_{M_t}\left(\log_0^c(f_D^{h,s,r})\right)\right\| \leq d\left\|\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right\|$. Similarly, $\|\log_0^c(Y_{M_t})\| \leq d\left\|\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right\|$ and $\|\log_0^c(Y_{i,M_t})\| \leq d\left\|\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right\|$, because $\left(\pi_{M_t}(\log_0^c(f_D^{h,s,r}))\right), Y_{M_t}, Y_{i,M_t} \in \left[-\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c}), \frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right]^d$.

Now, from the expression of $\mathcal{E}_{\pi_{M_t}}(f_D^{h,s,r})$, we get

$$\mathcal{E}_{\pi_{M_t}}(f_D^{h,s,r}) = \int_{\mathcal{X}\times\mathcal{Y}} \left\|\log_0^c(f_D^{h,s,r})_{M_t} - \log_0^c(y)_{M_t}\right\|^2 d\rho \leq 4d\left(\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right)^2.$$

Similarly, we get the same bound for the empirical errors, and thus we get,

$$\left\|\mathcal{E}_{\pi_{M_t}}\left(\pi_{M_t} f_D^{h,s,r}\right) - \mathcal{E}_{\pi_{M_t}}\left(\pi_{M_t} f_\rho\right)\right\| \leq 8d\left(\frac{1}{\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right)^2.$$

By putting $\alpha = \beta = 1$ and $\epsilon = t^{-\theta}$ in Lemma 11 we get,

$$\begin{aligned} &\left(\mathcal{E}_{\pi_{M_t}}\left(\pi_{M_t} f_D^{h,s,r}\right) - \mathcal{E}_{\pi_{M_t}}(f_\rho)\right) - \left(\mathcal{E}_{\pi_{M_t}}(\pi_{M_t} f_D^{h,s,r}) - \mathcal{E}_{\pi_{M_t}}(\pi_{M_t} f_\rho)\right) \\ &\leq 8dM_t^2\left(\frac{1}{M_t\sqrt{c}} \tanh^{-1}(M_t\sqrt{c})\right)^2 t^{-\theta}, \end{aligned} \tag{31}$$

holds with probability at most,

$$14 \sup_{X_1^t} \mathcal{N}_1\left(\left(\frac{t^{-\theta}}{2M_t}\right)^d \frac{1}{20}, \pi_{M_t} U, X_1^t\right) \times \exp\left(-\frac{t^{1-2d\theta}}{214 \times (3{M_t}^4)^d}\right),$$

where $U = \mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$. We now use Corollary 12 to derive that,

$$\sup_{X_1^t} \mathcal{N}_1 \left(\left(\frac{t^{-\theta}}{2M_t} \right)^d \frac{1}{20}, \pi_{M_t} U, X_1^t \right) \times \exp\left(- \frac{t^{1-2d\theta}}{214 \times (3{M_t}^4)^d} \right)$$
$$\leq \exp\left[A \log(B) \right] \times \exp\left(- \frac{t^{1-2d\theta}}{214 \times (3{M_t}^4)^d} \right), \text{ where}$$

$A := Ctd^2 \log(d), B := dt \left(2{M_t}' t^\theta\right)^{dL} \log\left(dt \left(2{M_t}' t^\theta\right)^{dL} \right)$, and ${M_t}' := \frac{1}{\sqrt{c}} \tanh^{-1}(M_t \sqrt{c})$

Therefore, by the conditions of Theorem 1, for $(h, s, r) \geq (2, 1, 4)$ we get

$$\lim_{t \to \infty} \sup_{X_1^t} \mathcal{N}_1 \left(\left(\frac{t^{-\theta}}{2M_t} \right)^d \frac{1}{20}, \pi_{M_t} U, X_1^t \right) \times \exp\left(- \frac{t^{1-2d\theta}}{214 \times (3{M_t}^4)^d} \right) = 0.$$

Finally, using the Borel-Cantelli Lemma, we get

$$\mathcal{E}_{\pi_{M_t}}(\pi_{M_t} f_D^{h,s,r}) - \mathcal{E}_{\pi_{M_t}, D}(\pi_{M_t} f_D^{h,s,r}) = 0$$

holds almost surely, completing the proof of Lemma 14. □

B.3 Step 3: Universal Approximations of Transformers:

The following lemma serves as the final component in establishing Theorem 1, which extends the universal approximation property of [61] to the hyperbolic setting. As we mentioned earlier, we will assume $(h, s, r) \geq (2, 1, 4)$ for the approximation property to hold.

Lemma 15. *For any given $\epsilon > 0$ and a function $g : \mathbb{D}_c^{d \times \lceil t/L \rceil} \to \mathbb{D}_c^{d \times \lceil t/L \rceil}$, there exists $f \in \mathcal{T}_{\mathcal{P},\mathcal{H}}^{hs,r}$ such that,*

$$\delta_2(f, g) := \left[\int_{\mathcal{X}} \| \log_0^c(f(x)) - \log_0^c(g(x)) \|_2^2 dx \right]^{1/2} < \epsilon. \tag{32}$$

Proof of Lemma 15. We define $h := \log_0^c \circ g \circ \exp_0^c$. Then h is a continuous map from $\mathbb{R}^{d \times t} \to \mathbb{R}^{d \times t}$. Now g is compactly supported and $\exp_0^c : \mathbb{R}^{d \times \lceil t/L \rceil} \to \mathbb{D}_c^{d \times \lceil t/L \rceil}$ is a global diffeomorphism. Thus, h is a continuous map from $\mathbb{R}^{d \times \lceil t/L \rceil} \to \mathbb{R}^{d \times \lceil t/L \rceil}$ with compact support. Now, by Theorem 3 in [61], we know that there exists a $b \in T_{\mathcal{P},\mathcal{H}}^{h,s,r}$ such that

$$\left[\int_Y \| h(y) - b(y) \|_2^2 dy \right]^{1/2} < \frac{\epsilon}{C}, \tag{33}$$

where $C = \sup_{x \in \mathcal{R} \subsetneq \mathbb{D}_c^{d \times \lceil t/L \rceil}} \| \log_0^c(x) \|$. We have $C < \infty$, since $\mathcal{R}$ is a compact subset of $\mathbb{D}_c^{d \times \lceil t/L \rceil}$ and $\log_0^c$ is a global diffeomorphism on $\mathcal{R}$. We will now construct the target function f from b. Define $f := \exp_0^c \circ b \circ \log_0^c$. Then $f \in \mathcal{T}_{\mathcal{P},\mathcal{H}}^{hs,r}$. We also note that

$$\begin{aligned}
\delta_2(f, g) &= \left[\int_{\mathcal{X}} \| \log_0^c(f(x)) - \log_0^c(g(x)) \|_2^2 dx \right]^{1/2} \\
&= \left[\int_{\mathcal{X}} \| h \circ \log_0^c(x) - b \circ \log_0^c(x) \|_2^2 dx \right]^{1/2} \\
&= \left[\int_{\mathcal{X}} \| (h - b) \circ (\log_0^c(x)) \|_2^2 dx \right]^{1/2} \\
&\leq C \left[\int_Y \| h(y) - b(y) \|_2^2 dy \right]^{1/2} \leq \epsilon,
\end{aligned}$$

where the last inequality is followed from Equation (33). □

B.4 Other Technical proofs

Proof of Theorem 1. Since we know that $\mathbb{E}[(\log_0^c(y))^2] < \infty$, by Lemma 10, for all $\epsilon > 0$ there exists $f \in \mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$ such that $\delta_2(f, f_\rho) < \epsilon$.

By the triangle inequality, we bound the following difference into a sum of 8 terms,

$$
\begin{aligned}
&\mathcal{E}(\pi_{M_t}(f_D^{h,s,r})) - \mathcal{E}(f_\rho) \\
&\leq \left[\epsilon\mathcal{E}\left(\pi_{M_t}(f_D^{h,s,r})\right) - (1+\epsilon)\mathcal{E}(\pi_{M_t}(f_D^{h,s,r}))\right] + \left[(1+\epsilon)\left(\mathcal{E}_{\pi_{M_t}}(\pi_M(f_D^{h,s,r}))\right) - \mathcal{E}_{\pi_{M_t},D}(\pi_{M_t}(f_D^{h,s,r}))\right] \\
&+ \left[(1+\epsilon)\left(\mathcal{E}_{\pi_{M_t},D}(\pi_{M_t}(f_D^{h,s,r})) - \mathcal{E}_{\pi_M,D}(f_D^{h,s,r})\right)\right] + \left[(1+\epsilon)(\mathcal{E}_{\pi_{M_t},D}(f_D^{h,s,r})) - (1+\epsilon)^2(\mathcal{E}_D(f_D^{h,s,r}))\right] \\
&+ \left[(1+\epsilon)^2\left(\mathcal{E}_D(f_D^{h,s,r}) - \mathcal{E}_D(f)\right)\right] + \left[(1+\epsilon)^2\left(\mathcal{E}_D(f) - \mathcal{E}(f)\right)\right] + \left[(1+\epsilon)^2\left(\mathcal{E}(f) - \mathcal{E}(f_\rho)\right)\right] \\
&+ \left[\left((1+\epsilon)^2 - 1\right)\mathcal{E}(f_\rho)\right] \\
&=: \sum_{i=1}^{8} A_i,
\end{aligned}
$$

where each of the A_i's represents a term in the summation. We will show that under the conditions of Theorem 1, each of the $A_i \to 0$. We note that, for each $q, w, \epsilon > 0$,

$$(q+w)^2 \leq (1+\epsilon)q^2 + (1+1/\epsilon)w^2. \tag{34}$$

We begin by showing that $A_1 \to 0$ as $t \to \infty$.

$$
\begin{aligned}
A_1 &= \epsilon\mathcal{E}\left(\pi_{M_t}(f_D^{h,s,r})\right) - (1+\epsilon)\mathcal{E}(\pi_{M_t}(f_D^{h,s,r})) \\
&= \int_{\mathcal{Z}} \|\pi_{M_t}(f_D^{h,s,r}(x)) - (\log_0^c(y_{M_t})) + (\log_0^c(y_{M_t})) - (\log_0^c(y))\|^2 d\rho \\
&- (1+\epsilon)\int_{\mathcal{Z}} \|\pi_{M_t}(\log_0^c(f_D^{h,s,r})) - (\log_0^c(y_{M_t}))\|^2 d\rho \\
&\leq (1+(1/\epsilon))\int_{\mathcal{Z}} \|\log_0^c(y) - \log_0^c(y_{M_t})\|^2 d\rho. \quad \text{[By (34)]}
\end{aligned}
$$

Since $\epsilon > 0$ is arbitrary, by condition (i) in Theorem 1, we have $A_1 \to 0$ as $t \to \infty$. Next, combining Lemma 15 and the conditions of Theorem 1, we get $A_2 \to 0$ as $t \to \infty$. The definition of the truncation operator results in

$$A_3 = \frac{1}{t}\sum_{i=1}^{t} \|\pi_{M_t}(\log_0^c(f_D^{h,s,r}(X_i))) - (\log_0^c(Y_{i,M_t}))\|^2 - \frac{1}{t}\sum_{i=1}^{t} \|\log_0^c(f_D^{h,s,r}(X_i)) - (\log_0^c(Y_{i,M_t}))\|^2 \leq 0.$$

Using the Strong Law of Large Numbers and the inequality (34), we deduce

$$A_4 \leq (1+\epsilon)(1+1/\epsilon)\frac{1}{t}\sum_{i=1}^{t} \|\log_0^c(Y_i) - \log_0^c(Y_{i,M_t})\|^2 \to (1+\epsilon)(1+1/\epsilon)\int_{\mathcal{Z}} \|\log_0^c(y) - \log_0^c(y_{M_t})\|^2 d\rho$$

as $t \to \infty$ almost surely. And the condition (i) in Theorem 1 implies $A_4 \to 0$ as $t \to \infty$. Since $f_D^{h,s,r}$ is obtained by minimizing the empirical risk, we have,

$$A_5 = (1+\epsilon)^2\left(\frac{1}{t}\sum_{i=1}^{t} \|\log_0^c(f_D^{h,s,r}(X_i)) - \log_0^c(Y_i)|^2\right) - (1+\epsilon)^2\left(\frac{1}{t}\sum_{i=1}^{t} \|\log_0^c(f(X_i)) - \log_0^c(Y_i)\|^2\right) \leq 0.$$

Once again, the strong law of large numbers directly indicates $A_6 \to 0$ almost surely as $t \to \infty$. We can write A_7 as $A_7 = (1+\epsilon)^2\|f - f\rho\|^2_{L^2_{\rho_\mathcal{X}}}$, which goes to 0 by Lemma 15. Finally, for A_8, we have

$$A_8 \leq ((1+\epsilon)^2 - 1)\int_{\mathcal{Z}} \|\log_0^c(f_\rho(x)) - \log_0^c(y)\|^2 d\rho = \epsilon(\epsilon+2)\int_{\mathcal{Z}} \|\log_0^c(f_\rho(x)) - \log_0^c(y)\|^2 d\rho,$$

which definitely goes to 0 as $\epsilon > 0$ is arbitrary and the integral is bounded. Summing up from A_1 to A_8, we have

$$\limsup_{t\to\infty} \mathcal{E}(\pi_{M_t}(f_D^{h,s,r})) - \mathcal{E}(f_\rho) \leq (1+\epsilon)^2\epsilon + \epsilon(2+\epsilon) \times \int_{\mathcal{Z}} \| \log_0^c(f_\rho(x)) - \log_0^c(y)\|^2 d\rho \tag{35}$$

holds almost surely. As $\epsilon > 0$ is arbitrary, we can write

$$\limsup_{m\to\infty} \mathcal{E}(\pi_{M_t}(f_D^{h,s,r})) - \mathcal{E}(f_\rho) = 0.$$

This completes the proof of Theorem 1. □

Lemma 16 (**Sample Complexity.**). *For any finite truncation parameter M, the empirical error converges to the universal error at a rate of $\mathcal{O}(t^{-1/2d})$ with probability at least $1 - C^* P \log(Q) \exp\left(-\frac{t^{1-2d\theta}}{214\times(3M^4)^d}\right)$, where P and Q are as defined in Theorem 1, and $C^* > 0$ is a constant.*

Proof of Lemma 16. Note that, in the proof of Lemma 8, we can substitute $\epsilon = m^{-\theta}, \theta \in (0, 1/2d)$. Additionally, the convergence proved in Lemma 9 was instrumental in proving Theorem 1, particularly in establishing Equation 35. Since the right-hand side of Equation (35) is linear in ϵ, and the minimal approximation error $\int_{\mathcal{Z}} \| \log_0^c(f_\rho(x)) - \log_0^c(y)\|^2, d\rho$ is finite, we obtain:

$$\mathcal{E}(f_t) - \mathcal{E}(f_\rho) = \mathcal{O}(m^{-1/2d}), \tag{36}$$

thereby completing our claim regarding the sample complexity of the hypothesis class $\mathcal{T}_{\mathcal{P},\mathcal{H}}^{h,s,r}$, where $f_t = \pi_{M_t} f_D^{h,s,r}$. □

Corollary 17 (**Consistency in Euclidean Regime**). *Given $\theta \in (0, 1/2d)$, the number of input tokens t, and input embedding dimension d, if the following conditions hold as $t \to \infty$*

(i) The sequence of the truncation parameters $M_t \to \infty$ and hence,

(ii) $t^{-\theta} M_t^2 \to 0$ as $t \to \infty$, and

(iii) $\frac{P\log(Q)}{t^{1-2\theta d}} \to 0$, where $P = M_t^4 d\log(d)$ and $Q = dt(2M_t t^\theta)^{dL} \log\left(dt\left(2M_t t^\theta\right)^{dL}\right)$,

then, $\pi_{M_t} \bar{f}_D^{h,s,r}$ is strongly universally consistent for all $(h, s, r) \geq (2, 1, 4)$.

Table 3: The details of the hyperparameters for the number of attention heads (h), size of the heads (s), and hidden dimension for the feed-forward network (r) are provided. Each triplet satisfies the criteria $(h, s, r) \geq (2, 1, 4)$ to justify the theoretical analysis through the simulations.

Hyperparameters	Squad	TweetQA	MNIST	CIFAR10	PATTERN	CLUSTER
h (# attention heads)	2	4	8	8	8	8
s (head size)	256	32	32	32	8	8
r (hidden dimension)	4	128	1024	1024	64	64

Table 4: The input embedding dimension and number of training tokens

Paremeters	Squad	TweetQA	MNIST	CIFAR10	PATTERN	CLUSTER
d(embedding dimension)	128	128	64	64	3	7
t(training tokens)	$12M$	$100k$	$60k$	$50k$	$1.2M$	$1.2M$

C Additional Experiments and Details

C.1 Additional Results

We have mentioned the performance on the six datasets in the Table 2. The vanilla Transformers equipped with hyperbolic components achieved impressive performances for the higher curvatures compared to their Euclidean versions. The results strongly support our theoretical analysis across the diverse sets of tokens.

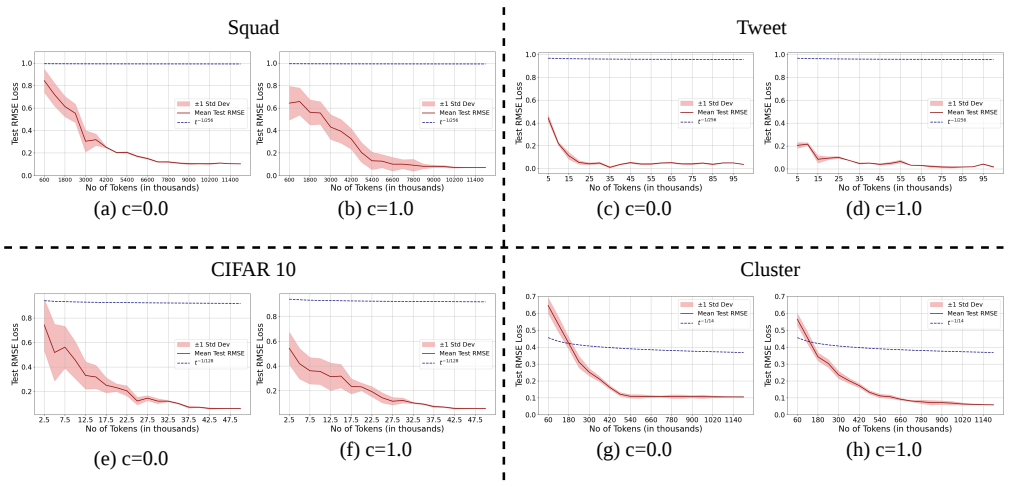
C.2 Additional Experiments on Sample Complexity

We further evaluate the theoretical sample complexity bound established in Lemma 16 on the remaining four datasets, namely SQuAD, Tweet, CIFAR-10, and CLUSTER. These datasets span language, image, and graph domains, thereby allowing us to examine the robustness of the derived rate across different data modalities and input structures. For each dataset, we divide the available training samples into 20 equally sized partitions and progressively increase the effective training sample size across these partitions. At each sample size, the Transformer is trained using the corresponding subset of the training data, and the resulting Test RMSE is recorded to characterize the empirical convergence behavior as a function of the number of training samples.

We then compare the empirical Test RMSE curves against the corresponding theoretical sample complexity bounds derived in Lemma 16. Across all four datasets, the empirical curves remain below the respective theoretical bound curves as the number of training samples increases. Moreover, the observed decrease in Test RMSE is consistent with the predicted asymptotic behavior of the theoretical rate. These results provide empirical evidence that the derived sample complexity bound captures the dependence of the estimation error on the available number of training samples across substantially different data modalities. In particular, the consistent behavior observed for language, image, and graph datasets suggests that the theoretical characterization is not restricted to a particular data domain, but instead provides a unified description of the convergence behavior of the proposed Transformer framework. Overall, these experiments further corroborate the theoretical sample complexity result established in Lemma 16.

C.3 Hyperparameter Details

We provide the details of (h, s, r) for each of the datasets in Table 3. The parameters of the Transformer are optimized with Adam. For graph transforms, we considered Laplacian vectors as our primary positional encodings. In case of Graph Transformers, BatchNorm is employed instead of Layernorm to obtain the best performance. For ViT, the learnable positional encodings are employed, and for language Transformers, we employed sinusoidal PEs. Necessary dropouts are also applied. Additionally, LayerNorm and BatchNorm layers are also incorporated to attain faster convergence.



(a) c=0.0 (b) c=1.0 (c) c=0.0 (d) c=1.0

(e) c=0.0 (f) c=1.0 (g) c=0.0 (h) c=1.0

Figure 6: Validating theoretical sample complexity bounds $\mathcal{O}(t^{-1/2d})$ [Lemma 16] on Squad, Tweet, CIFAR10 and Cluster datasets for two different curvatures

C.4 Computational Resources

The entire experiment is executed on a single NVIDIA RTX GeForce 24GB GPU. The system is equipped with a 2TB hard drive, 64GB RAM, and Ubuntu OS.