

# HYPERBOLIC FINE-TUNING FOR LARGE LANGUAGE MODELS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Large language models (LLMs) have demonstrated remarkable performance on various tasks. However, it remains an open question whether the default Euclidean space is the most suitable choice for embedding tokens in LLMs. In this study, we first investigate the non-Euclidean characteristics of LLMs. Our findings reveal that token frequency follows a power-law distribution, with high-frequency tokens clustering near the origin and low-frequency tokens positioned farther away. Additionally, token embeddings exhibit a high degree of hyperbolicity, indicating a latent tree-like structure in the embedding space. Building on the observation, we propose to efficiently fine-tune LLMs in hyperbolic space to better exploit the underlying complex structures. However, we found that this fine-tuning in hyperbolic space cannot be achieved with naive application of exponential and logarithmic maps, when the embedding and weight matrices both reside in Euclidean space. To address this technique issue, we introduce a new method called hyperbolic low-rank efficient fine-tuning, HypLoRA, that performs low-rank adaptation directly on the hyperbolic manifold, avoiding the cancellation effect caused by the exponential and logarithmic maps, thus preserving the hyperbolic modeling capabilities. Through extensive experiments, we demonstrate that HypLoRA significantly enhances the performance of LLMs on reasoning tasks, particularly for complex reasoning problems. In particular, HypLoRA improves the performance in the complex AQuA dataset by up to 13.0%, showcasing its effectiveness in handling complex reasoning challenges.

## 1 INTRODUCTION

Large language models (LLMs) such as GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and Gemma (Gemma Team, 2024) have demonstrated remarkable capabilities in understanding and generating human-like text (Qin et al., 2023; Shen et al., 2024). Despite their impressive capabilities, these models often rely on Euclidean geometry for learning text representations, which may not always be the complex, hierarchical nature of real-world data structures (Bronstein et al., 2017; Bachmann et al., 2020). For example, in language, words are often organized into categories that reflect varying levels of abstraction. These relationships naturally form a tree-like structure, where general or abstract concepts, such as "fruits," sit at the top of the hierarchy, while more specific or concrete terms, like "apples" or "bananas," reside at the lower levels. Representing such structures effectively is crucial for understanding the semantics of language in LLMs.

Recent advancements suggest that non-Euclidean geometries, particularly hyperbolic spaces, offer promising alternatives for modeling hierarchical data. Hyperbolic space, distinguished by its negative curvature, is especially well-suited for representing tree-like data due to its exponential volume growth, enabling efficient embeddings of hierarchies (Nickel & Kiela, 2017; 2018; Ganea et al., 2018a; Khruikov et al., 2020; Cetin et al., 2022). However, a significant research gap remains: existing works have not attempted to study LLM embeddings in the context of non-Euclidean geometry.

**Proposed Analysis Framework** In this work, we first delve deep into how LLMs interact with token embeddings and explore to what extent these embeddings exhibit non-Euclidean characteristics. We approach this from both a global and local perspective. At the global level, we analyze the overall distribution of tokens by frequency, examining how these frequency maps are distributed across the embedding space. At the local level, we measure the hyperbolicity (Borassi et al., 2015; Kennedy

et al., 2013) of the metric space spanned by each input prompt, where the embedding hyperbolicity serves as a proxy to assess the similarity of the underlying embedding structure to a tree structure.

Our analysis in Section 4 reveals several key insights. First, token frequency follows a power-law distribution, as shown in Figure 2 (left). This distribution, where a small set of tokens appears frequently while most are rare, suggests an implicit hierarchy similar to a branching tree (Krioukov et al., 2010). High-frequency tokens (abstract concepts) tend to be located near the origin of the embedding space, while low-frequency tokens (specific terms) are farther away, as depicted in Figure 2 (right) and Table 1. Furthermore, our investigation of hyperbolicity ( $\delta$  values) in Table 2 demonstrates that LLM token embeddings exhibit significant tree-like properties.

Based on our findings above, a natural consideration is to develop hyperbolic LLMs that explicitly incorporate hyperbolic inductive bias, as shown in Figure 1. However, training LLMs

from scratch can be resource-intensive (Loshchilov & Hutter, 2017; Rajbhandari et al., 2020). As a more resource-efficient alternative, we propose to build the first low-rank adaptation method in hyperbolic space. This approach is particularly advantageous given that existing LLMs are all Euclidean, and not all downstream tasks require hyperbolic geometry in their fine-tuning. Through employing hyperbolic adapters for specific tasks on an Euclidean foundation model, we can leverage the benefits of both geometries while maintaining computational efficiency.

**Challenges** Adapting LLMs in non-Euclidean embedding spaces with classic techniques, *i.e.* applying exponential and logarithmic maps with tangent space (Chami et al., 2019; Ganea et al., 2018b; Yang et al., 2022c) for weight adaptation is problematic in this case. This approach fails to fully capture the hyperbolic geometry, as the exponential and logarithmic maps are mutually inverse and can be canceled with consecutive operations. Consequently, the inherent properties of the hyperbolic space are not effectively preserved, limiting the potential benefits of incorporating non-Euclidean geometries into the adaptation process.

**Hyperbolic Fine-tuning** To address this limitation, we introduce HypLoRA to operate low-rank adaptation directly on the hyperbolic manifold without transformation to the tangent space, thus preserving hyperbolic modeling capabilities and counteracting the reduction<sup>1</sup>. HypLoRA integrates hyperbolic geometry into existing LLMs, introducing implicitly high-order interaction and considering the token hierarchies, enabling them to benefit from hyperbolic characteristics while minimizing additional computational costs.

To summarize, our main contributions are threefold: (1) We conduct a comprehensive investigation into the hyperbolic characteristics of token embeddings in LLMs, revealing their inherent tree-like structure and strong hyperbolic properties. (2) We propose HypLoRA, a parameter-efficient fine-tuning method that integrates hyperbolic geometry into LLMs while preserving hyperbolic modeling capabilities. We show that HypLoRA better understands complex reasoning tasks by implicitly incorporating high-order interactions and token norms, achieving improvements of up to 13% on the challenging AQuA dataset. Our work opens new avenues for exploring the role of geometry in LLMs and provides insights for developing geometrically informed models for reasoning tasks.

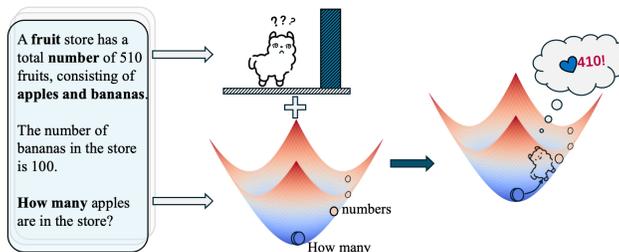


Figure 1: Illustration of a reasoning task with hyperbolic geometry on LLMs. The figure shows that a fruit store has 510 fruits (apples and bananas), and knowing that there are 100 bananas, we determine the number of apples to be 410. In token embedding, frequent and abstract tokens (like, "fruit, how many, and numbers") are represented closer to the origin, while specific and less frequent tokens (like "apples, bananas, numbers, 510, 110") appear further away, creating a tree-like structure. By this structure, hyperbolic space enables LLMs to map hierarchical relationships more efficiently, preserving the inherent structure of tokens, understanding the semantic meaning, and facilitating accurate reasoning.

<sup>1</sup>Code is available at <https://anonymous.4open.science/r/HypLLMs-DD7A>

## 2 RELATED WORK

**Hyperbolic Representation Learning and Deep Learning** Hyperbolic geometry has been successfully applied to various neural network architectures and models (Yang et al., 2022b; Mettes et al., 2023; Peng et al., 2021), including shallow hyperbolic neural networks (Ganea et al., 2018a;b; Chen et al., 2021; Shimizu et al., 2020), hyperbolic CNNs (Bdeir et al., 2023; van Spengler et al., 2023), and hyperbolic attention networks or Transformers (Gulcehre et al., 2018; Chen et al., 2021; Shimizu et al., 2020; Yang et al., 2024). These models leverage the inductive biases of hyperbolic geometry to achieve remarkable performance on various tasks and applications (Chami et al., 2019; Yang et al., 2022a; Sun et al., 2021; Khurlov et al., 2020; Cetin et al., 2022; Weng et al., 2021; Xiong et al., 2022; Yang et al., 2021). However, training LLMs from scratch remains computationally expensive (Kochurov et al., 2020; Smith, 2014). The computational complexity increases further when considering Riemannian optimization (Kochurov et al., 2020; Smith, 2014; Bécigneul & Ganea, 2018) and additional hyperbolic operations, like Möbius addition.

**Parameter Efficient Fine Tuning (PEFT) and LoRAs** Fine-tuning LLMs (Foundation, 2022; 2023; Touvron et al., 2023) for downstream tasks poses significant challenges due to their massive number of parameters. To address this issue, PEFT methods have been proposed, which aim to train a small subset of parameters while achieving better performance compared to full fine-tuning. PEFT methods can be broadly categorized into prompt-based methods (Lester et al., 2021; Li & Liang, 2021; Qin et al., 2021), adapter-based methods (Houlsby et al., 2019; Zhu et al., 2021), and reparameterization-based methods (Hu et al., 2021; Aghajanyan et al., 2020; Edalati et al., 2022). Among these, LoRA (Hu et al., 2021) as the reparameterization-based method, has gained significant attention due to its simplicity, effectiveness, and compatibility with existing model architectures. Variants of LoRA, such as LoRA+ (Hayou et al., 2024), DoRA (Liu et al., 2024), AdaLoRA (Zhang et al., 2023), have been proposed to improve its performance and efficiency. Recent research has also investigated ensembles of multiple LoRAs (Wang et al., 2023; Ren et al., 2024), and quantization techniques (Dettmers et al., 2024; Xu et al., 2023; Li et al., 2023). Despite these advances, existing methods operate within Euclidean space, ignoring the underlying structure represented by LLMs. The proposed method is as a foundational algorithm, potentially combined with various LoRA variants, to exploit their complementary strengths and achieve superior performance.

## 3 PRELIMINARY

This section introduces the concepts utilized in our study, including the LoRA adapter, the Lorentz model of hyperbolic geometry, hyperbolic linear transformations, and the concept of hyperbolicity.

**LoRA Adapter** The LoRA adapter offers an efficient approach for modifying large LLMs with minimal computational overhead. Instead of retraining the entire model, LoRA focuses on adjusting specific components within the model’s architecture to transform an input  $\mathbf{x}$  into an output  $\mathbf{z}$ . In practice, LoRA targets the weight matrices found in each Transformer layer of an LLM. Typically, the weight  $W$  of the Transformer, which resides in the dimensions  $\mathbb{R}^{d \times k}$ , is adapted through a low-rank approximation. This is achieved by introducing an additional term,  $\Delta W$ , to the original weight matrix:

$$\mathbf{z} = W_{\text{LoRA}}(\mathbf{x}) = W\mathbf{x} + \Delta W\mathbf{x} = W\mathbf{x} + B A \mathbf{x}. \quad (1)$$

Here,  $B \in \mathbb{R}^{d \times r}$  and  $A \in \mathbb{R}^{r \times k}$  represent two smaller, learnable matrices where  $r$ —the rank of these matrices—is significantly less than either  $d$  or  $k$ . This design choice ensures that  $r \ll \min(d, k)$ , thereby reducing the complexity of the model adaptation. During the fine-tuning process, only the matrices  $A$  and  $B$  are adjusted, while the pre-existing weights  $W$  are kept frozen. This method significantly decreases the number of parameters that need to be trained, from  $dk$  to  $(d + k)r$ , enhancing the efficiency of the fine-tuning process. As a result, LoRA enables the targeted adaptation of LLMs, allowing them to transform an input  $\mathbf{x}$  into an output  $\mathbf{z}$  while maintaining high performance and adapting to new tasks or datasets with a fraction of the computational resources typically required.

**Hyperbolic Geometry** Unlike the flat Euclidean geometry, hyperbolic geometry is characterized by a constant negative curvature. We utilize the Lorentz model, also known as the hyperboloid model, for our study due to its ability to effectively capture hierarchical structures and maintain numerical stability (Nickel & Kiela, 2018; Chen et al., 2021). The Lorentz model in  $n$  dimensions with curvature

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

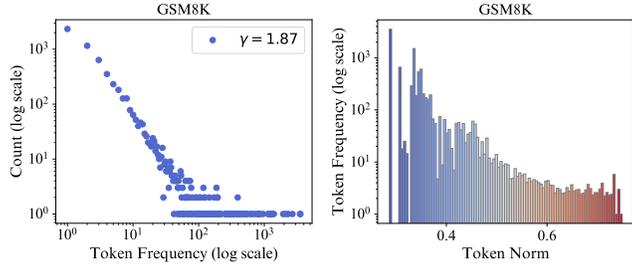


Figure 2: Token frequency distribution (left) and token frequency vs. norm (bottom row) of GSM8K dataset in LLaMA3. The top row shows the power-law distribution of token frequencies with the decay rate ( $\gamma$ ) annotated for each dataset. The bottom row illustrates the relationship between token frequency and token norm, binned and colored by frequency, where higher token norms correspond to lower frequencies. For more data illustration, please refer to Appendix A.

$-1/K (K > 0)$  is defined as:

$$\mathcal{L}_K^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -K, x_0 > 0\}, \tag{2}$$

where  $\langle \cdot, \cdot \rangle_{\mathcal{L}}$  is the Lorentzian inner product, given by:  $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i$ .

**Tangent Space** In the Lorentz model  $\mathcal{L}_K^n$ , the tangent space at a point  $\mathbf{x}$  is denoted as  $\mathcal{T}_x \mathcal{L}_K^n$ . It is defined as the set of all vectors  $\mathbf{u}$  that are orthogonal to  $\mathbf{x}$  under the Lorentzian inner product:

$$\mathcal{T}_x \mathcal{L}_K^n := \{\mathbf{u} \in \mathbb{R}^{n+1} : \langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}} = 0\}. \tag{3}$$

To facilitate projection between the hyperboloid and its tangent spaces, we utilize two critical mappings: the exponential and logarithmic maps. The *exponential map* at  $\mathbf{x}$ , denoted  $\exp_{\mathbf{x}}^K$ , projects a vector from the tangent space  $\mathcal{T}_x \mathcal{L}_K^n$  back onto the hyperboloid. Conversely, the *logarithmic map*, denoted  $\log_{\mathbf{x}}^K$ , maps a point on the hyperboloid to the tangent space at  $\mathbf{x}$ . The relevant formulas are given in Appendix C.

## 4 INVESTIGATION

In this section, we present an in-depth investigation of token embeddings in LLMs from both global and local perspectives. Our goal is to uncover the geometric structures underlying pretrained token representations, specifically examining the global distribution of token frequencies and their spatial arrangement, as well as the local hyperbolicity of token embeddings across various datasets.

### 4.1 GLOBAL TOKEN STATISTICS

We begin by investigating the global distribution of token frequencies in the context of arithmetic reasoning datasets, focusing on datasets such as GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). We also provide a broader analysis across different types of datasets and LLMs in Appendix A.

Figure 2 (left) presents the distribution of token frequencies, with a power-law exponent of approximately  $\gamma \approx 1.9$ , as estimated by the Powerlaw Package (Alstott et al., 2014). In such distributions, the exponent  $\gamma$  controls how quickly token frequencies decline: smaller values of  $\gamma$  (closer to 1) indicate a more gradual decay where frequent tokens dominate, while larger values signify a sharper decline, with most tokens being rare.

This power-law behavior aligns with the hierarchical nature of language. High-frequency tokens often correspond to more abstract or general concepts, while low-frequency tokens represent specific or rare terms. This distribution naturally suggests a hierarchical organization of the token space, where general concepts serve as the "roots" and specific terms "branch out" as we move through the hierarchy.

To better understand the relationship between token frequency and their spatial arrangement within the embedding space, we calculate the average token frequency as a function of their distance from the origin. The results are shown in Figure 2 (right), indicating that more frequent tokens tend to have smaller norms and vice versa. Table 1 provides representative with different norm ranges within the embedding space. Tokens with smaller norms (ranging from 0.3 to 0.4) include common function words like "to," "in," and "that," which tend to occur frequently in language. As the norm increases, the tokens become more specific, with ranges from 0.4 to 0.5 including terms like "how," "much," and "time," and further up, with norms between 0.5 and 0.6, featuring more concrete nouns like "animals," "fruit," and "numbers." Tokens with the highest norms, between 0.6 and 0.7, are even more detailed, referring to items such as "dog," "apple," and "dollars."

These findings suggest that the spatial organization of token embeddings reflects the inherent hierarchical relationships in language, supporting the hypothesis that LLMs exhibit a tree-like structure in their token embeddings, with spatial positioning aligned with token frequency and specificity.

#### 4.2 $\delta$ -HYPERBOLICITY OF TOKEN EMBEDDINGS

To rigorously quantify the hierarchical nature of token embeddings, we examine the  $\delta$ -hyperbolicity of space spanned by the token embedding.  $\delta$ -Hyperbolicity, introduced by Gromov (Gromov, 1987), is a measure that captures the degree to which a metric space deviates from an exact tree structure. Lower values of  $\delta$  imply a space more similar to a perfect tree, while higher values indicate deviation from a tree-like structure. A brief explanation of  $\delta$ -hyperbolicity can be found on Wikipedia<sup>2</sup>.

We compute  $\delta$ -hyperbolicity using the four-point condition, which compares the Gromov products between any four points  $a$ ,  $b$ ,  $c$ , and  $w$  in the metric space. Specifically, the hyperbolicity is defined as:

$$[a, c]_w \geq \min([a, b]_w, [b, c]_w) - \delta, \quad (4)$$

where the Gromov product  $[a, b]_w$  is:

$$[a, b]_w = \frac{1}{2}(d(a, w) + d(b, w) - d(a, b)). \quad (5)$$

To measure the hyperbolicity of token embeddings, we apply this algorithm to various open-source LLMs. Following the methodologies proposed by Khruikov et al. (Khruikov et al., 2020) and Cetin et al. (Cetin et al., 2022), we estimate  $\delta$ -hyperbolicity using the efficient algorithm introduced by Fournier et al. (Fournier et al., 2015). To ensure scale invariance, we normalize  $\delta$  by the diameter of the embedding space,  $\text{diam}(X)$ , yielding a relative measure:  $\delta_{rel} = \frac{2\delta}{\text{diam}(X)}$ . This relative measure ranges from 0 to 1, with values closer to 0 indicating a highly hyperbolic (tree-like) structure, and values near 1 indicating a non-hyperbolic, flat structure. Following previous works (Khruikov et al., 2020), we employ Euclidean distance as a measure of the shortest distance. To further validate the correctness of this approach, we generate a series of random graphs with predefined hyperbolicity, embed them using a graph neural network (GNN), and then compute the hyperbolicity in Euclidean space. Details of this process are provided in Appendix B. Our experiments reveal a positive correlation between the hyperbolicity of the embeddings and the original graphs. Consequently, we utilize this method as a proxy for estimating the hyperbolicity of token embeddings.

In our analysis, we calculate hyperbolicity at the prompt level, treating each token within a prompt as a point in the metric space spanned by the embeddings. By averaging the hyperbolicity across all prompts, we assess the overall hyperbolic structure of token embeddings in each dataset. Our results, as shown in Table 2, reveal that token embeddings exhibit significant hyperbolicity, suggesting that the embedding space has a strong tree-like structure. This observation further corroborates our findings from the global token statistics, where the arrangement of tokens in the embedding space mirrors hierarchical relationships seen in language data.

<sup>2</sup>[https://en.wikipedia.org/wiki/Hyperbolic\\_metric\\_space](https://en.wikipedia.org/wiki/Hyperbolic_metric_space)

Table 2: Comparison of  $\delta$ -Hyperbolicity across various metric spaces and datasets. The left table provides reference values for baseline metric spaces, allowing for a clearer interpretation of hyperbolicity in the analyzed datasets in the right table.

Metric Space	Hyperbolicity( $\delta$ )	Hyperbolicity( $\delta$ )	MAWPS	SVAMP	GSM8K	AQuA
Sphere Space	$0.99 \pm 0.01$	LLaMA-7B	$0.08 \pm 0.02$	$0.09 \pm 0.01$	$0.10 \pm 0.01$	$0.10 \pm 0.01$
Random Graph	$0.62 \pm 0.34$	LLaMA-13B	$0.08 \pm 0.01$	$0.09 \pm 0.01$	$0.09 \pm 0.01$	$0.10 \pm 0.01$
PubMed Graph	$0.40 \pm 0.45$	Gemma-7B	$0.11 \pm 0.01$	$0.11 \pm 0.01$	$0.11 \pm 0.01$	$0.12 \pm 0.01$
Scale-free Graph	0.00	LLaMA3-8B	$0.06 \pm 0.01$	$0.07 \pm 0.01$	$0.07 \pm 0.01$	$0.08 \pm 0.01$
Tree Graph	0.00	Average	$0.08 \pm 0.01$	$0.09 \pm 0.01$	$0.09 \pm 0.01$	$0.10 \pm 0.01$

**Conclusion of investigation** Through these analyses, we demonstrate that token embeddings in LLMs exhibit a power-law frequency distribution and significant hyperbolicity, both of which reflect a tree-like hierarchical structure. This understanding not only sheds light on the geometric nature of token embeddings but also motivates the development of methods that better capture and preserve these underlying geometric properties.

## 5 HYPERBOLIC FINE-TUNING FOR LLMs

The core technique in the LoRA adapter involves linear transformations. One of the primary methods for implementing linear transformations on the Lorentz model of hyperbolic geometry (Ganea et al., 2018b; Chami et al., 2019) is based on the tangent space when considering the learnable weights are in Euclidean. Given a hyperbolic vector  $\mathbf{x}^H$  and a transformation matrix  $W$ , this method first maps  $\mathbf{x}^H$  to the tangent space at a local reference point, typically the origin, using the logarithmic map. The matrix  $W$  is then applied within this tangent space, resulting in:

$$W \otimes \mathbf{x}^H = \exp(W \log_{\mathbf{o}}^K(\mathbf{x}^H)). \quad (6)$$

**Technical Challenge** However, the input from LLMs and the transformation results are in Euclidean space, we need to apply an additional exponential map and a logarithmic map on the basis of Equation (1) to align the Euclidean representation. This leads to the expression:

$$\begin{aligned} \mathbf{z}^E &= W_{\text{LoRA}}(\mathbf{x}^E) = W\mathbf{x}^E + \Delta W\mathbf{x}^E \\ &= W\mathbf{x}^E + \log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\underbrace{BA \log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\mathbf{x}^E))}_{\text{Transformation on } \mathbf{x}^E})) \\ &= W\mathbf{x}^E + BA\mathbf{x}^E, \end{aligned} \quad (7)$$

which simplifies back to the original LoRA, rendering the method ineffective for our purposes.

**Direct Lorentz Low-rank Transformation (LLR)** To address this challenge, we perform low-rank adaptation directly on the hyperbolic manifold without utilizing tangent space:

$$\begin{aligned} \mathbf{z}^E &= W_{\text{LoRA}}(\mathbf{x}^E) = W\mathbf{x}^E + \Delta W\mathbf{x}^E \\ &= W\mathbf{x}^E + \log_{\mathbf{o}}^K(\underbrace{\text{LLR}(BA, \exp_{\mathbf{o}}^K(\mathbf{x}^E))}_{\text{Transformation on } \mathbf{x}^H}), \end{aligned} \quad (8)$$

where **LLR** represents the direct Lorentz Low-Rank Transformation which operate the hyperbolic representation  $\mathbf{x}^H$  directly,

$$\text{LLR}(BA, \mathbf{x}^H) = (\sqrt{\|B\mathbf{y}_*^H\|_2^2 + K}, B\mathbf{y}_*^H), \text{ where } \mathbf{y}^H = (\sqrt{\|A\mathbf{x}_*^H\|_2^2 + K}, A\mathbf{x}_*^H), \quad (9)$$

We consider two transformations in our design, with  $\mathbf{u}$  representing both  $\mathbf{x}$  and  $\mathbf{y}$ : (1)  $\mathbf{u}_*^H = \mathbf{u}_s$ . (2)  $\mathbf{u}_*^H = \mathbf{u}$ . The first transformation only modifies the space-like dimension in special relativity, akin to a Lorentz rotation. The second transformation affects both time-like and space-like dimensions, similar to a Lorentz boost. In both cases, it can be verified that  $\text{LLR}(BA, \mathbf{x}^H) \in \mathcal{L}^n$ . The linear

transformation is inspired by hyperbolic neural networks (Chen et al., 2021; Yang et al., 2024; Dai et al., 2021). For efficient integration with LLMs, the transformation removes normalization and non-linear activation term in (Chen et al., 2021), varying curvatures in (Yang et al., 2024), and orthogonal constraints in (Dai et al., 2021). Our main contribution lies in applying hyperbolic low-rank adaptation for LLMs, while the specific linear transformation itself is flexible—other transformations on the manifold could also be compatible with our approach.

In summary, our proposed method, HypLoRA, initially uses the exponential map to project the original Euclidean representation into hyperbolic space, applies a low-rank Lorentz transformation, and then employs the logarithmic map to revert to Euclidean space. By adapting in the hyperbolic domain, HypLoRA captures more complex hierarchical relationships than traditional Euclidean-based methods, as detailed in Proposition 5.1. Additionally, the low-rank nature of the adaptation matrices  $A$  and  $B$  promotes parameter efficiency, making HypLoRA well-suited for LLMs.

**Time Complexity** HypLoRA has similar theoretical time complexity as the Euclidean LoRA, which is  $\mathcal{O}(r \cdot (d + k))$ , where  $d$  and  $k$  represent the input and output dimensions, respectively. However, in practical implementation, HypLoRA introduces additional computations due to the logarithmic and exponential maps. These additional operations, nevertheless, can be completed within  $\mathcal{O}(N)$  where the  $N$  is the number of input tokens.

**Proposition 5.1.** *Let  $\mathbf{x}$  represent the input token embeddings, with  $\|\mathbf{x}\|$  denoting their norms. HypLoRA modifies the query and key updates by introducing higher-order terms that depend on  $\|\mathbf{x}\|$ . This dependence enables HypLoRA to capture the hierarchical relationships in token embeddings. As a result, HypLoRA aligns with the intrinsic geometry of token embeddings.*

## 5.1 EXPERIMENTAL SETTINGS

**Dataset** The experimental setup closely follows the methodology in (Hu et al., 2023). The fine-tuning training set is composed of data from GSM8K (Cobbe et al., 2021), MAWPS, MAWPS-single (Koncel-Kedziorski et al., 2016), and 1,000 examples from AQuA (Ling et al., 2017). To further enhance reasoning capabilities, step-by-step rationales generated by ChatGPT are incorporated into the training samples, as done in (Hu et al., 2023). This results in a dataset of 10K math reasoning samples, named Math-10K, for training purposes. The test datasets include GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). While the same datasets are used for training, there is no overlap between the training and test sets.

**Model Comparison** We include the LLaMA-7B and LLaMA-13B base models, as discussed in (Hu et al., 2023), along with the recently released Gemma-7B and LLaMA3-8B models, which are fine-tuned using LoRA for comparison. For fine-tuning methods, we evaluate several techniques, including Prefix-Tuning (Li & Liang, 2021), Series Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021), and Parallel Adapter (He et al., 2021). Additionally, we compare with DoRA (Liu et al., 2024), a recent competitive method.

**Implementation Details** The exponential map transforms the original input space using an exponential operator, as also noted in (Desai et al., 2023). To prevent numerical overflow, we first apply L2 normalization to the input before using the exponential map in Equation (8), then rescale it with a learnable norm scaling factor. The curvature for our proposed HypLoRA is treated as a hyperparameter, with values searched from the set  $\{0.1, 0.5, 1.0, 2.0\}$ . Following the procedure in (Chami et al., 2019), to correctly use the exponential map, we append a zero to the beginning of the input vector  $\mathbf{x}$ , forming  $\mathbf{x}^E$ . After applying the logarithmic map, the output vector  $\mathbf{z}$  will have an additional dimension with a zero value. To maintain consistency with the original input space, we remove this extra dimension from  $\mathbf{z}$ . It is important to note that the final results are micro-averaged across datasets, which contain varying numbers of questions, such as 1,319 in GSM8K and 238 in MAWPS. In micro-averaging, each prompt is treated equally. For the LoRA implementation, we insert it into both the Multi-head Attention and MLP layers of the base model. All experiments are conducted on a single GPU, using either the A40 (40G) or A100 (80G).

## 5.2 EXPERIMENTAL RESULTS

Table 3 summarizes our key experimental outcomes, highlighting both the baseline model performance and the improvements from incorporating adapters. Since our experimental setup and dataset selection

Table 3: Accuracy comparison of various LLMs using PEFT methods on arithmetic reasoning tasks. Results marked with an asterisk (\*) are sourced from Hu et al. (Hu et al., 2023). (†) denotes our reproduced results on LoRA. The LoRA results for LLaMA3-8B and Gemma-7B are derived using the hyperparameters specified in the same study. The percentage following each dataset indicates the proportion of prompts relative to the total number of inference prompts. M.AVG represents the micro-average accuracy. OOT denotes evaluations exceeding 24 hours on an A100 GPU, while None refers to the base model without any PEFT method applied. NA stands for Not Applicable.

Model	PEFT Method	MAWPS(8.5%)	SVAMP(35.6%)	GSM8K(46.9%)	AQuA(9.0%)	MAVG
GPT-3.5	None	<b>87.4</b>	<b>69.9</b>	<b>56.4</b>	<b>38.9</b>	<b>62.3</b>
LLaMA-7B	None	51.7	32.4	15.7	16.9	24.8
	Prefix*	63.4	38.1	24.4	14.2	31.7
	Series*	77.7	52.3	33.3	15.0	42.2
	Parallel*	82.4	49.6	35.3	18.1	42.8
	LoRA*	79.0	52.1	37.5	18.9	44.6
	LoRA†	81.9	48.2	38.3	18.5	43.7
	DoRA	80.0	48.8	39.0	16.4	43.9
	<b>HypLoRA (Ours)</b>	79.0	49.1	39.1	20.5	44.4
LLaMA-13B	None	65.5	37.5	32.4	15.0	35.5
	Prefix*	66.8	41.4	31.1	15.7	36.4
	Series*	78.6	50.8	44.0	22.0	47.4
	Parallel*	81.1	55.7	43.3	20.5	48.9
	LoRA*	83.6	54.6	47.5	18.5	50.5
	LoRA†	83.5	54.7	48.5	18.5	51.0
	DoRA	83.0	54.6	OOT	18.9	NA
	<b>HypLoRA (Ours)</b>	83.2	54.8	49.0	21.5	51.5
Gemma-7B	None	76.5	60.4	38.4	25.2	48.3
	LoRA	91.6	76.2	66.3	28.9	68.6
	DoRA	91.7	75.9	65.4	27.7	68.0
	<b>HypLoRA (Ours)</b>	91.5	78.7	69.5	32.7	71.3
LLaMA3-8B	None	79.8	50.0	54.7	21.0	52.1
	LoRA	<b>92.7</b>	78.9	70.8	30.4	71.9
	DoRA	92.4	79.3	71.3	33.1	72.5
	<b>HypLoRA (Ours)</b>	91.6	<b>80.5</b>	<b>74.0</b>	<b>34.2</b>	<b>74.2</b>

closely align with those used by Hu et al. (Hu et al., 2023), we reference their results directly. For the new base models, like Gemma-7B and LLaMA3-8B, we maintained the same training strategy for consistency. Each experiment was run three times, and we reported the average results. We have the following findings:

**Overall Performance of HypLoRA and Challenging Datasets** The accuracy from GPT-3.5 indicates that GSM8K and AQuA are among the more challenging datasets in this evaluation. The performance of the models and the fine-tuned results is strongly related to the difficulty level of the datasets. It is as complex problems require more complex reasoning and a better understanding of the underlying structure of the problem. Nonetheless, HypLoRA consistently outperforms the baseline methods across various base models. Notably, the overall performance improvement reaches up to 2.3% on LLaMA3-8B and 3.9% on Gemma-7B against the best competitors. In addition, HypLoRA method excels on these complex datasets, AQuA and GSM8K, where it achieves an improvement of up to 13.0% on the AQuA dataset and 4.8% on the GSM8K dataset with the Gemma-7B model. This significant gain reflects the advantage of introducing hyperbolic geometry, as its inherent geometric properties make it better suited for capturing complex, hierarchical structured data. The analysis in Appendix D demonstrates that HypLoRA implicitly introduces high-order interaction terms and highlights higher-order terms proportional to the token norm, correlating with more specific tokens in the token hierarchy. This enables the model to focus on more tokens and better comprehend complex relationships. Consequently, HypLoRA effectively leverages the hierarchical and hyperbolic structure of the data, resulting in improved performance on challenging reasoning tasks.

**Performance with DoRA and on MAWPS Dataset** The computational complexity of DoRA leads to timeouts in evaluations, such as for GSM8K on LLaMA-13B. Despite this, HypLoRA consistently performs as well or better than DoRA in completed evaluations, offering comparable results with much lower computational overhead. On the MAWPS dataset, the performance improvements of HypLoRA are less pronounced compared to other datasets, and in some instances, it falls below the baseline results. This may be attributed to using the same curvature during fine-tuning, which might not be suitable for all datasets. To address this, future work will focus on prompt-adaptive curvature

432 techniques. Despite this limitation, HypLoRA has demonstrated significant improvements over the  
 433 base model across the majority of datasets evaluated.

### 434 5.3 ABLATION STUDY AND PARAMETER ANALYSIS

435  
 436  
 437  
 438  
**Ablation Study** We use  
 439 the tangent-space method  
 440 described in Equation (7)  
 441 as a basis for conducting  
 442 an ablation study. The  
 443 primary difference between  
 444 the tangent-space method  
 445 and the proposed HypLoRA  
 446 lies in the approach used for  
 447 the Low-rank Transformat-  
 448 ion. Through this compar-  
 449 ison, we can determine the effectiveness and benefits of the direct Lorentz Low-rank approach.  
 450 Furthermore, compared to the Euclidean LoRA, both Equation (7) and the proposed HypLoRA  
 451 incorporates an additional rescaling operation, as discussed in Section 5.1. Considering that the  
 452 tangent-space method can be reduced to the general LoRA form, it can be viewed as an additional  
 453 rescaling operation combined with the vanilla LoRA. By making these comparisons, we can evaluate  
 454 the effectiveness of normal rescaling.

455 Table 4 presents our results. HypLoRA (I) denotes the first transformation (Lorentz rotation) on  
 456 space-like dimensions and HypLoRA (II) denotes the second transformation (Lorentz boost) on  
 457 the whole dimensions. We observe that the tangent space method shows improvement over the  
 458 original LoRA, which is expected since the rescaling step introduces more flexibility. Due to the  
 459 exponential effects in hyperbolic geometry, this rescaling step is necessary. Comparing the results of  
 460 HypLoRA with the tangent space method, we can see the significant impact of introducing hyperbolic  
 461 geometry, with the main improvements attributed to this incorporation. We also observe that these  
 462 two transformations obtain similar performances.

463 **The Impact of Curvature on Performance** Curvature in  
 464 hyperbolic space is a key hyperparameter in HypLoRA, di-  
 465 rectly affecting its capacity to model underlying structures  
 466 and geometries. To evaluate its impact, we experimented  
 467 with different curvature values on the Gemma 7B model, as  
 468 shown in Table 5, where curvature is defined as  $-1/K$ . Our  
 469 results demonstrate that overall model performance remains  
 470 relatively stable across various curvature settings. Notably,  
 471 determining the optimal curvature value was straightforward,  
 472 with a final value of 1.0 proving to be the best. In future work, we will consider exploring the data-  
 473 informed curvature method to make the fine-tuning more adaptive.

474 **Inference Efficiency** In Section 5, we analyze the time complexity of our approach, which remains  
 475 consistent with that of LoRA. However, during actual inference, HypLoRA incurs additional com-  
 476 putational overhead due to operations such as the exponential and logarithmic mappings. These  
 477 operations introduce some additional runtime, particularly for larger models. The GPU hours for  
 478 inference on four datasets are presented in Figure 3. Despite this overhead, our method demonstrates  
 479 improved efficiency when compared to the previous competitive model, DoRA. Notably, HypLoRA  
 480 still outperforms DoRA in terms of both runtime and overall efficiency. In future work, we plan  
 481 to investigate more efficient approaches to further reduce this computational cost, with the goal of  
 482 minimizing the impact of these transformations on overall inference time.

483 **Case Study** The introduction of hyperbolic space allows the model to capture more token information  
 484 and accurately comprehend the semantics of the prompt, leading to more effective solutions for  
 485 complex reasoning tasks. Table 6 provides examples to demonstrate HypLoRA’s capabilities. In  
 this case, LoRA fails to grasp the true nature of the computational task, while HypLoRA correctly  
 interprets the problem by recognizing the three-week timeframe and appropriately distributing the

Table 4: Ablation Study

Model	Methods	MAWPS	SVAMP	GSM8K	AQuA	M.AVG
<b>Gemma 7B</b>	Tangent	91.2	77.9	67.6	30.5	69.9
	HypLoRA (I)	93.2	78.3	68.5	33.2	70.9
	HypLoRA (II)	91.5	78.7	69.5	32.7	<b>71.3</b>
<b>LLaMA 8B</b>	Tangent	91.2	79.2	72.3	30.3	72.6
	HypLoRA (I)	91.6	80.2	74.1	33.8	74.1
	HypLoRA (II)	91.6	80.5	74.0	34.2	<b>74.2</b>

Table 5: Results for varying of K

K	0.1	0.5	1.0	2.0
MAWPS	89.8	91.7	91.5	90.8
SVAMP	78.1	77.3	78.7	78.6
GSM8K	68.5	67.5	69.5	68.5
AQuA	31.9	34.3	32.7	31.1

required time. This highlights the superior ability of semantics understanding and the ability to handle multi-step reasoning.

Furthermore, the introduction of hyperbolic space enhances the model’s ability to comprehend hierarchical relationships between tokens, providing better separation of long-tail tokens for prediction. This is due to the exponential growth properties of hyperbolic space, which fit these tokens more effectively and yield more accurate reasoning outcomes. Additional case studies are provided in Appendix E.

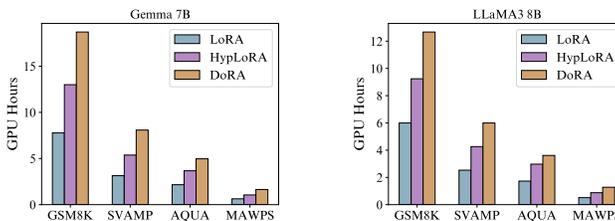


Figure 3: Illustration of GPU (A100) hours during inference

Table 6: Example problem with solutions from LoRA and HypLoRA

<b>Question</b>	It takes 240 minutes of walking to break in a new pair of shoes. Jason wants to try out for the track team in three weeks. If he can walk 4 days a week to break in the new shoes, how long does he have to spend walking each day? <b>Answer:</b> 20.0 minutes
<b>LoRA</b>	To find out how long Jason has to spend walking each day, we need to divide the total number of minutes it takes to break in the shoes by the number of days he has to do it. Jason has 240 minutes divided by 4 days per week. This results in 60 minutes per day. <b>Answer:</b> 60.
<b>HypLoRA</b>	First, we need to find how many minutes Jason needs to walk in total to break in the new shoes: 240 minutes. Next, we find how many minutes Jason needs to walk each week over three weeks: 240 minutes divided by 3 weeks equals 80 minutes per week. Then, to find how many minutes he needs to walk each day: 80 minutes per week divided by 4 days equals 20 minutes per day. <b>Answer:</b> 20.

## 6 CONCLUSION

In this study, we investigated the non-Euclidean properties of token embeddings in LLMs. Our analysis token embedding has strong hyperbolic characteristics. Building on these findings, we proposed HypLoRA, a hyperbolic low-rank adaptation method that incorporates hyperbolic geometry into the fine-tuning process of LLMs. Extensive experiments showed that HypLoRA significantly improves LLM performance on arithmetic reasoning tasks, particularly on complex datasets. By leveraging the hyperbolic structure of the data, HypLoRA enhances the model’s ability to capture and utilize intricate relationships, leading to better reasoning capabilities.

**Limitation and Future Work** In this study, we employed a consistent curvature across all prompts during fine-tuning, which simplified the implementation and enhanced efficiency. However, this uniform approach may not be optimal when applied to different datasets simultaneously. Our future work will explore more adaptive fine-tuning techniques that can better accommodate the unique characteristics of different prompts. Additionally, due to the computational overhead introduced by the exponential and logarithmic maps, this is inevitable when transitioning from the original Euclidean space to a hyperbolic space. We will explore more efficient methods to reduce this computational cost in future work.

Despite these challenges, our research provides a thorough examination of token embedding distributions from a non-Euclidean perspective and offers valuable insights. The fine-tuning method we proposed holds significant potential for advancing geometrically inspired models, contributing to the ongoing development of more effective LLMs.

**Reproducibility Statement** Due to the inherent randomness in large model inference, answers on individual prompts may slightly different, but overall results are consistent. We averaged results over multiple runs. For LoRA, we rerun LoRA on LLaMA-7B/13B, though performance on SVAMP with LLaMA-7B was lower than reported. All DoRA experiments were run on A100 GPUs due to long inference times on A40, with LLaMA-13B still exceeding 24 hours.

## REFERENCES

- 540  
541  
542 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,  
543 Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical  
544 Report. *arXiv preprint arXiv:2303.08774*, 2023.
- 545 Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the  
546 effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.
- 547  
548 Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: a python package for analysis of heavy-tailed  
549 distributions. *PloS one*, 9(1):e85777, 2014.
- 550 Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional  
551 networks. In *International conference on machine learning*, pp. 486–496. PMLR, 2020.
- 552  
553 Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Hyperbolic geometry in computer vision:  
554 A novel framework for convolutional neural networks. *arXiv preprint arXiv:2303.15919*, 2023.
- 555 Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv  
556 preprint arXiv:1810.00760*, 2018.
- 557  
558 Michele Borassi, Alessandro Chessa, and Guido Caldarelli. Hyperbolicity measures democracy in  
559 real-world networks. *Physical Review E*, 92(3):032812, 2015.
- 560  
561 Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric  
562 deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42,  
563 2017.
- 564 Edoardo Cetin, Benjamin Chamberlain, Michael Bronstein, and Jonathan J Hunt. Hyperbolic deep  
565 reinforcement learning. *arXiv preprint arXiv:2210.01542*, 2022.
- 566  
567 Ines Chami, Zitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural  
568 networks. *Advances in neural information processing systems*, 32, 2019.
- 569  
570 Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou.  
571 Fully hyperbolic neural networks. *arXiv preprint arXiv:2105.14686*, 2021.
- 572 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,  
573 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve  
574 math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 575  
576 Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional  
577 network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*,  
578 pp. 154–163, 2021.
- 579 Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam.  
580 Hyperbolic Image-Text Representations. In *Proceedings of the International Conference on  
581 Machine Learning*, 2023.
- 582  
583 Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning  
584 of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.
- 585 Ali Edalati, Marzieh Tahaei, Ivan Kobzyev, Vahid Partovi Nia, James J Clark, and Mehdi  
586 Rezagholizadeh. Krona: Parameter efficient tuning with kronecker adapter. *arXiv preprint  
587 arXiv:2212.10650*, 2022.
- 588  
589 OpenAI Foundation. Introducing chatgpt. <https://openai.com/index/chatgpt>, Novem-  
590 ber 2022.
- 591  
592 OpenAI Foundation. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- 593  
Hervé Fournier, Anas Ismail, and Antoine Vigneron. Computing the gromov hyperbolicity of a  
discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015.

- 594 Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning  
595 hierarchical embeddings. In *International Conference on Machine Learning*, pp. 1646–1655.  
596 PMLR, 2018a.
- 597 Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in*  
598 *neural information processing systems*, 31, 2018b.
- 600 Google Deepmind Gemma Team. Gemma: Open models based on gemini research and technology.  
601 *arXiv preprint arXiv:2403.08295*, 2024.
- 602
- 603 Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.
- 604
- 605 Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz  
606 Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention  
607 networks. *arXiv preprint arXiv:1805.09786*, 2018.
- 608 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics,  
609 and function using networkx. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman (eds.),  
610 *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pp. 11–15. Pasadena, CA USA,  
611 2008.
- 612 Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models.  
613 *arXiv preprint arXiv:2402.12354*, 2024.
- 614
- 615 Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a  
616 unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.
- 617
- 618 Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe,  
619 Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for  
620 nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.
- 621 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,  
622 and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint*  
623 *arXiv:2106.09685*, 2021.
- 624
- 625 Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing,  
626 and Soujanya Poria. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large  
627 language models. *arXiv preprint arXiv:2304.01933*, 2023.
- 628
- 629 W Sean Kennedy, Onuttom Narayan, and Iraj Saniee. On the hyperbolicity of large-scale networks.  
630 *arXiv preprint arXiv:1307.0031*, 2013.
- 631
- 632 Valentin Khruikov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky.  
633 Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision*  
634 *and pattern recognition*, pp. 6418–6428, 2020.
- 635
- 636 Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch.  
637 *arXiv preprint arXiv:2005.02819*, 2020.
- 638
- 639 Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps:  
640 A math word problem repository. In *Proceedings of the 2016 conference of the north american*  
641 *chapter of the association for computational linguistics: human language technologies*, pp. 1152–  
642 1157, 2016.
- 643
- 644 Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná.  
645 Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft*  
646 *Matter Physics*, 82(3):036106, 2010.
- 647
- 648 Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt  
649 tuning. *arXiv preprint arXiv:2104.08691*, 2021.
- 650
- 651 Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv*  
652 *preprint arXiv:2101.00190*, 2021.

- 648 Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo  
649 Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint*  
650 *arXiv:2310.08659*, 2023.
- 651 Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale genera-  
652 tion: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*,  
653 2017.
- 654 Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-  
655 Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint*  
656 *arXiv:2402.09353*, 2024.
- 657 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*  
658 *arXiv:1711.05101*, 2017.
- 659 Pascal Mettes, Mina Ghadimi Atigh, Martin Keller-Ressel, Jeffrey Gu, and Serena Yeung. Hyperbolic  
660 deep learning in computer vision: A survey. *arXiv preprint arXiv:2305.06611*, 2023.
- 661 Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations.  
662 In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6338–6347, 2017.
- 663 Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of  
664 hyperbolic geometry. In *International Conference on Machine Learning (ICML)*, pp. 3779–3788,  
665 2018.
- 666 Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math  
667 word problems? *arXiv preprint arXiv:2103.07191*, 2021.
- 668 Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic  
669 deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,  
670 2021.
- 671 Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi  
672 Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint*  
673 *arXiv:2302.06476*, 2023.
- 674 Yujia Qin, Xiaozhi Wang, Yusheng Su, Yankai Lin, Ning Ding, Jing Yi, Weize Chen, Zhiyuan Liu,  
675 Juanzi Li, Lei Hou, et al. Exploring universal intrinsic task subspace via prompt tuning. *arXiv*  
676 *preprint arXiv:2110.07867*, 2021.
- 677 Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations  
678 toward training trillion parameter models. In *SC20: International Conference for High Performance*  
679 *Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.
- 680 Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke,  
681 Zhumin Chen, and Jiahuan Pei. Mini-ensemble low-rank adapters for parameter-efficient fine-  
682 tuning. *arXiv preprint arXiv:2402.17263*, 2024.
- 683 Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad.  
684 Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.
- 685 Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hug-  
686 gingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. *Advances in Neural*  
687 *Information Processing Systems*, 36, 2024.
- 688 Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint*  
689 *arXiv:2006.08210*, 2020.
- 690 Steven Thomas Smith. Optimization techniques on riemannian manifolds. *arXiv preprint*  
691 *arXiv:1407.5965*, 2014.
- 692 Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. HGCF: Hyperbolic  
693 graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference*  
694 *(WWW)*, pp. 593–601, 2021.

- 702 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée  
703 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and  
704 efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 705 Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincaré resnet. In *Proceedings of the*  
706 *IEEE/CVF International Conference on Computer Vision*, pp. 5419–5428, 2023.
- 707 Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-  
708 tuning. *arXiv preprint arXiv:2310.00035*, 2023.
- 709 Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery  
710 of the long-tail in instance segmentation using hierarchical self-supervision. In *Proceedings of the*  
711 *IEEE/CVF conference on computer vision and pattern recognition*, pp. 2603–2612, 2021.
- 712 Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. Hyperbolic embedding inference  
713 for structured multi-label prediction. *Advances in Neural Information Processing Systems*, 35:  
714 33016–33028, 2022.
- 715 Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen,  
716 Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language  
717 models. *arXiv preprint arXiv:2309.14717*, 2023.
- 718 Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time  
719 temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings*  
720 *of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1975–1985,  
721 2021.
- 722 Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. Hicf: Hyperbolic informative  
723 collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge*  
724 *Discovery and Data Mining (KDD)*, pp. 2212–2221, 2022a.
- 725 Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. Hyperbolic  
726 graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852*,  
727 2022b.
- 728 Menglin Yang, Min Zhou, Hui Xiong, and Irwin King. Hyperbolic temporal network embedding.  
729 *IEEE Transactions on Knowledge and Data Engineering*, 2022c.
- 730 Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer:  
731 Exploring efficient transformer fully in hyperbolic space. *arXiv preprint arXiv:2407.01290*, 2024.
- 732 Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo  
733 Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International*  
734 *Conference on Learning Representations*, 2023.
- 735 Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference  
736 adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 2021.

## 743 A INVESTIGATION ON MORE DATASETS

744 In the main text, we focused on token distribution for GSM8K datasets. Here, we provide more token  
745 distribution for the AQUA and MAWPS mathematical reasoning datasets. Besides, we extend this  
746 analysis to include common sense reasoning datasets, specifically OpenBookQA and WinoGrande.  
747 These results are shown in Figure 4. The findings align with the conclusions drawn in the main text.  
748

## 749 B HYPERBOLICITY ON DIFFERENT METRIC SPACES

750 Table 2 presents the hyperbolicity values in both continuous (i.e., Sphere Space) and discrete metric  
751 spaces (i.e., Tree Graph, Scale-free Graph and Random Graph). We employ a consistent processing  
752 method, akin to the one mentioned in Section (4) for embedding spaces. Specifically, we sample  
753 1000 4-tuples, compute the delta value for each, and then take the maximum value.  
754  
755

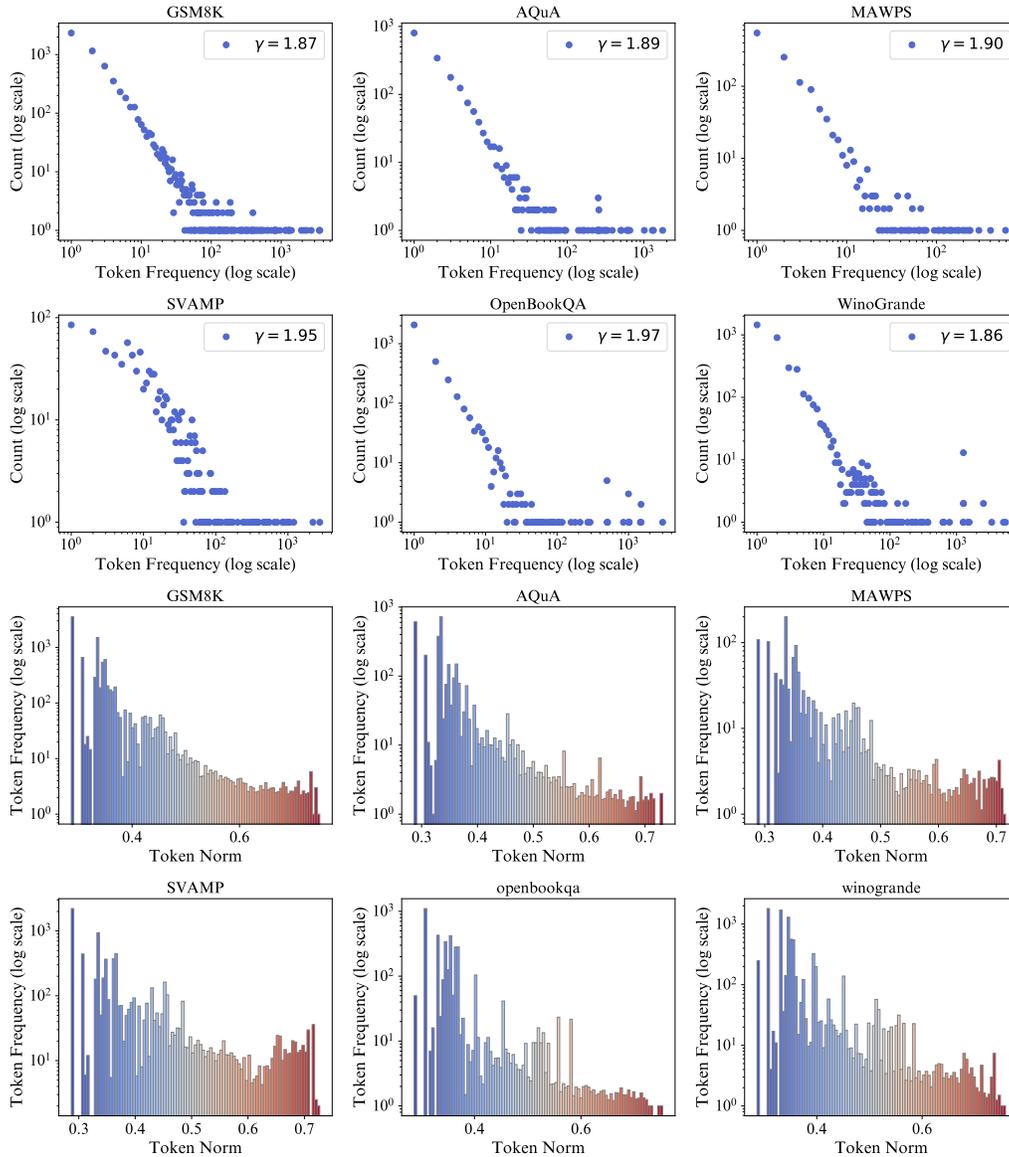


Figure 4: Token frequency distribution (top two rows) and token frequency vs. norm (bottom two rows) across different datasets in LLaMA3. The top two row show the power-law distribution of token frequencies with the decay rate ( $\gamma$ ) annotated for each dataset. The bottom two rows illustrates the relationship between token frequency and token norm, binned and colored by frequency, where higher token norms correspond to lower frequencies.

810 For the sphere space, we use a two-dimensional model and  
 811 calculate hyperbolicity based on their respective geodesic  
 812 distances. The PubMed graph is sourced from Sen et  
 813 al. (Sen et al., 2008). The tree graph and dense graph are  
 814 generated using NetworkX (Hagberg et al., 2008). For  
 815 these graphs, we first remove isolated nodes before per-  
 816 forming our calculations in a consistent manner. The short-  
 817 est path distance on the graph is used as the distance mea-  
 818 sure, analogous to the concept of geodesics in continuous  
 819 spaces.

820 In this study, we utilize the Euclidean distance to com-  
 821 pute the hyperbolicity of token embeddings, following the  
 822 approach proposed by Khruikov et al. (2020). To further  
 823 validate the correctness of this method, we embed graphs  
 824 with varying degrees of hyperbolicity into Euclidean space using a graph neural network (GNN)  
 825 model and compute hyperbolicity based on the distances between embeddings. The results, presented  
 826 in Figure 5, indicate a positive correlation between the hyperbolicity of the original graphs and  
 827 that of the embeddings, although the values do not exactly coincide. Building on this observed  
 828 relationship, we calculate the hyperbolicity of token embeddings as a proxy for estimating their  
 829 underlying geometric structure. In this context, lower hyperbolicity values suggest a more tree-like  
 830 geometric configuration.

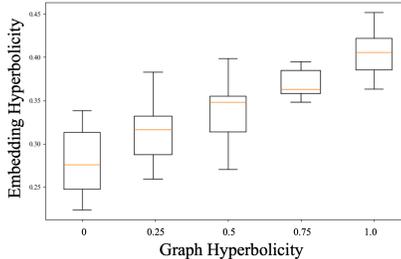


Figure 5: Correlation between graph and embedding hyperbolicity.

## 831 C EXPONENTIAL AND LOGARITHMIC MAP

833 The exponential and logarithmic maps serve as essential tools for projection between the local tangent  
 834 space and the hyperbolic space. Consider a point  $\mathbf{x} \in \mathcal{L}_K^n$  and a tangent vector  $\mathbf{u} \in \mathcal{T}_x \mathcal{L}_K^n$ . The  
 835 exponential map, denoted as  $\exp_x^K : \mathcal{T}_x \mathcal{L}_K^n \rightarrow \mathcal{L}_K^n$ , assigns to  $\mathbf{u}$  the point  $\exp_x^K(\mathbf{u}) := \gamma(1)$ , where  
 836  $\gamma$  represents the unique geodesic that satisfies the initial conditions  $\gamma(0) = \mathbf{x}$  and  $\dot{\gamma}(0) = \mathbf{u}$ . The  
 837 exponential map can be explicitly expressed as follows:

$$838 \exp_x^K(\mathbf{u}) = \cosh\left(\frac{\|\mathbf{u}\|_{\mathcal{L}}}{\sqrt{K}}\right) \mathbf{x} + \sqrt{K} \sinh\left(\frac{\|\mathbf{u}\|_{\mathcal{L}}}{\sqrt{K}}\right) \frac{\mathbf{u}}{\|\mathbf{u}\|_{\mathcal{L}}}, \quad (10)$$

840 where  $\cosh$  and  $\sinh$  represent the hyperbolic cosine and sine functions, respectively, and  $\|\mathbf{u}\|_{\mathcal{L}}$   
 841 denotes the norm of the tangent vector  $\mathbf{u}$  in the tangent space.

843 The logarithmic map  $\log_u^K(\mathbf{x}) : \mathcal{L}_K^n \rightarrow \mathcal{T}_u \mathcal{L}_K^n$  plays an inverse role. It is defined by the equation:

$$844 \log_u^K(\mathbf{x}) = \frac{\cosh^{-1}\left(-\frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}}\right)}{\sinh\left(\cosh^{-1}\left(-\frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}}\right)\right)} \left(\mathbf{x} + \frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}} \mathbf{u}\right). \quad (11)$$

847 The exponential and logarithmic maps establish a bijective projection between the tangent space  
 848 and hyperbolic space. Notably,  $\log_x^K(\exp_x^K(\mathbf{u})) = \mathbf{u}$  and  $\exp_u^K(\log_u^K(\mathbf{x})) = \mathbf{x}$ . **Consequently,**  
 849 **Equation (7) will cancel out the hyperbolic operations.** In addition, these operations are typically  
 850 defined locally. However, in the context of hyperbolic representation and deep learning, for efficient  
 851 computation, existing works usually use the origin point  $\mathbf{o} := \{\sqrt{K}, 0, \dots, 0\} \in \mathcal{L}_K^n$  as a common  
 852 reference point.

## 854 D TRANSFORMATION ANALYSIS

856 *Proof.* Let  $\mathbf{x} \in \mathbb{R}^n$  be an input token embedding, and  $A, B \in \mathbb{R}^{n \times r}$  be low-rank matrices with rank  
 857  $r \ll n$ . Consider the hyperbolic space  $\mathbb{H}^n$  with curvature  $C = -\frac{1}{R^2}$ , where  $R > 0$  is the radius of  
 858 curvature.

### 859 (1) Exponential Map at the Origin ( $\mathbf{o}$ ):

860 For a tangent vector  $\mathbf{v} \in T_{\mathbf{o}} \mathbb{H}^n$ :

$$861 \exp_{\mathbf{o}}(\mathbf{v}) = \left( \cosh\left(\frac{\|\mathbf{v}\|}{R}\right), \sinh\left(\frac{\|\mathbf{v}\|}{R}\right) \frac{\mathbf{v}}{\|\mathbf{v}\|} \right) \quad (12)$$

**(2) Logarithmic Map at the Origin (o):**

For  $\mathbf{x}^H = (x_0, \mathbf{x}_{\text{space}}) \in \mathbb{H}^n$ :

$$\log_{\mathbf{o}}(\mathbf{x}^H) = R \cdot \operatorname{arcosh}\left(\frac{x_0}{R}\right) \frac{\mathbf{x}_{\text{space}}}{\sqrt{x_0^2 - R^2}} \quad (13)$$

**(3) Applying Low-Rank Transformations:** For simplicity, we analysis about the transformation on space-like coordinates *First Transformation*:

$$\begin{aligned} \mathbf{y}_{\text{space}}^H &= A^\top \mathbf{x}_{\text{space}}^H, \\ y_0^H &= \sqrt{R^2 + \|\mathbf{y}_{\text{space}}^H\|^2}. \end{aligned} \quad (14)$$

*Second Transformation*:

$$\begin{aligned} \mathbf{z}_{\text{space}}^H &= B^\top \mathbf{y}_{\text{space}}^H, \\ z_0^H &= \sqrt{R^2 + \|\mathbf{z}_{\text{space}}^H\|^2}. \end{aligned} \quad (15)$$

**(4) Mapping Back to Euclidean Space:**

The update to the query vector is:

$$\Delta Q^{\text{Hyp}} = R \cdot \operatorname{arcosh}\left(\frac{z_0^H}{R}\right) \frac{\mathbf{z}_{\text{space}}^H}{\|\mathbf{z}_{\text{space}}^H\|} \quad (16)$$

**(5) Approximations Incorporating Token Norms:**

From the investigation, we know that token norms  $\|\mathbf{x}\|$  are correlated with their specificity in the hierarchical structure: tokens with larger norms represent more specific concepts.

For small  $\frac{\|\mathbf{x}\|}{R}$ , we use the Taylor series expansions:

- $\cosh\left(\frac{z}{R}\right) \approx 1 + \frac{z^2}{2R^2}$ .
- $\sinh\left(\frac{z}{R}\right) \approx \frac{z}{R} + \frac{z^3}{6R^3}$ .

Therefore, the spatial component after the exponential map is:

$$\mathbf{x}_{\text{space}}^H \approx \frac{\mathbf{x}}{R} + \frac{\|\mathbf{x}\|^2}{6R^3} \mathbf{x} \quad (17)$$

**(6) Applying the Transformations:**

*First Transformation*:

$$\mathbf{y}_{\text{space}}^H \approx \frac{A^\top \mathbf{x}}{R} + \frac{\|\mathbf{x}\|^2}{6R^3} A^\top \mathbf{x} \quad (18)$$

*Second Transformation*:

$$\mathbf{z}_{\text{space}}^H \approx \frac{(BA)\mathbf{x}}{R} + \frac{\|\mathbf{x}\|^2}{6R^3} (BA)\mathbf{x} \quad (19)$$

**(7) Approximating the Logarithmic Map:**

918 Compute  $z_0^H$ :

$$919 \quad z_0^H = \sqrt{R^2 + \|\mathbf{z}_{\text{space}}^H\|^2} \approx R + \frac{\|\mathbf{z}_{\text{space}}^H\|^2}{2R} \quad (20)$$

923 Compute  $\text{arcosh}\left(\frac{z_0^H}{R}\right)$ :

924 For small  $\delta = \frac{\|\mathbf{z}_{\text{space}}^H\|^2}{2R^2}$ :

$$925 \quad \text{arcosh}(1 + \delta) \approx \sqrt{2\delta} = \frac{\|\mathbf{z}_{\text{space}}^H\|}{R} \quad (21)$$

932 Final Expression for  $\Delta Q^{\text{Hyp}}$ :

$$933 \quad \Delta Q^{\text{Hyp}} \approx \mathbf{z}_{\text{space}}^H. \quad (22)$$

### 936 (8) Comparing HypLoRA and LoRA Updates:

937 HypLoRA Update:

$$938 \quad \Delta Q^{\text{Hyp}} \approx \frac{(BA)\mathbf{x}}{R} + \frac{\|\mathbf{x}\|^2}{6R^3}(BA)\mathbf{x}. \quad (23)$$

942 LoRA Update:

$$943 \quad \Delta Q^{\text{LoRA}} = \frac{(BA)\mathbf{x}}{R}. \quad (24)$$

947 Difference:

$$948 \quad \Delta Q^{\text{Hyp}} - \Delta Q^{\text{LoRA}} = \frac{\|\mathbf{x}\|^2}{6R^3}(BA)\mathbf{x}. \quad (25)$$

### 952 (9) Impact of Token Norms on Higher-Order Terms:

953 Since  $\|\mathbf{x}\|$  reflects the specificity of the token in the hierarchical structure (larger norms correspond to  
954 more specific tokens), the higher-order term  $\frac{\|\mathbf{x}\|^2}{6R^3}(BA)\mathbf{x}$  becomes significant for tokens representing  
955 specific concepts.  
956

### 957 (10) Impact on Attention Scores:

958 The HypLoRA attention scores are computed as:

$$959 \quad \text{Scores}_{\text{HypLoRA}} = \frac{(Q^{\text{orig}} + \Delta Q^{\text{Hyp}})(K^{\text{orig}} + \Delta K^{\text{Hyp}})^{\top}}{\sqrt{d_k}}. \quad (26)$$

964 where  $\Delta K^{\text{Hyp}}$  is derived similarly.

965 The difference in attention scores includes higher-order terms dependent on  $\|\mathbf{x}\|^2$ :

$$966 \quad \Delta \text{Scores} = \text{Scores}_{\text{HypLoRA}} - \text{Scores}_{\text{LoRA}}. \quad (27)$$

969 These higher-order terms allow HypLoRA to capture more complex, hierarchical relationships,  
970 particularly for tokens with larger norms (more specific tokens).  
971

□

**Remark D.1. Alignment with Token Hierarchy:** *The higher-order terms in HypLoRA’s updates are proportional to  $\|\mathbf{x}\|^2$ , which, according to our investigation, correlates with the specificity of tokens in the hierarchical structure. This means HypLoRA places greater emphasis on more specific tokens, enhancing its ability to model detailed relationships.*

**Role of Curvature  $C$ :** *The curvature scales the higher-order corrections, and in spaces with larger negative curvature (smaller  $R$ ), these terms become more pronounced, aligning with the hyperbolic nature of token embeddings.*

**Effectiveness of HypLoRA:** *By incorporating these higher-order terms, HypLoRA effectively leverages the inherent hierarchical and hyperbolic structure of token embeddings, leading to improved performance, especially the problems that require complex reasoning. This is also the reason why the proposed method has a better performance on more challenging datasets.*

## E CASE STUDY

In this section, we present examples to demonstrate the advantages of incorporating HypLoRA. As the examples show in Table 7 and 8, HypLoRA consistently provides more accurate reasoning compared to LoRA. For instance, in Example 2, LoRA incorrectly calculates the sum of the test scores as 380 instead of the correct total of 400, which leads to an incorrect average. This mistake highlights a limitation of large models like LoRA, which do not perform arithmetic operations as precisely as computers but rather approximate probability distributions. While the values 380 and 400 may seem close in Euclidean space, the hyperbolic geometry utilized by HypLoRA more effectively handles such variations, particularly in "tail" distributions, allowing for greater separation between close values and thus leading to more accurate reasoning. A similar issue arises in Example 3, where LoRA incorrectly computes  $11,600 - 9,800$ , resulting in a wrong conclusion about the load capacity of Gary’s truck. HypLoRA, on the other hand, correctly performs the calculation and arrives at the accurate answer. These examples demonstrate how the hyperbolic space used by HypLoRA enables better handling of complex reasoning tasks, improving both accuracy and reliability in such scenarios.

Table 7: Case Study 2

<b>Question</b>	To have the car for the weekend, Wilson’s report card needs to show that he received 80 or higher in his math class. His math grade is calculated by the average of 5 test scores. On those tests, he received: 65, 94, 81, 86, and 74. What is his current math grade? <b>Answer:</b> 80.0
<b>LoRA</b>	To find Wilson’s current math grade, we need to take the average of his 5 test scores. First, we add up all 5 test scores: $65 + 94 + 81 + 86 + 74 = 380$ . Then, we divide the sum by the number of tests: $380 \div 5 = 76$ . Therefore, Wilson’s current math grade is 76. <b>Answer: 76.</b>
<b>HypLoRA</b>	To find Wilson’s current math grade, we need to take the average of his 5 test scores. To do this, we add up the scores and divide by 5: $65 + 94 + 81 + 86 + 74 = 400$ . Then, we divide by 5: $400 \div 5 = 80$ . Therefore, Wilson’s current math grade is 80. <b>Answer: 80.</b>

1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079

Table 8: Case Study 3

<b>Question</b>	Gissela, Gordy, and Gary are truck drivers. Gissela has a truck large enough to haul 4,000 pounds of gravel. Gordy’s truck can haul 800 pounds more than Gissela’s truck. When Gary brings his truck and joins Gissela and Gordy, the three trucks combined can haul a total of 11,600 pounds of gravel. How many pounds of gravel can Gary’s truck carry? <b>Answer: 2,800 pounds</b>
<b>LoRA</b>	Let’s start by assigning variables to represent the amount of gravel each truck can haul: Gissela’s truck hauls 4,000 pounds. Gordy’s truck hauls $4,000 + 800 = 4,800$ pounds. Now, we set up an equation for the total amount of gravel the three trucks can haul: $4,000 + 4,800 + x = 11,600$ , where $x$ is the amount Gary’s truck hauls. Simplifying the equation: $9,800 + x = 11,600$ , we subtract 9,800 from both sides to get $x = 11,600 - 9,800 = 1,800$ . Therefore, Gary’s truck hauls 1,800 pounds. <b>Answer: 1,800.</b>
<b>HypLoRA</b>	Gissela’s truck can haul 4,000 pounds of gravel. Gordy’s truck can haul 800 more pounds than Gissela’s truck, so it can haul 4,800 pounds of gravel. Together, Gissela and Gordy’s trucks can haul 8,800 pounds of gravel. If the three trucks combined can haul 11,600 pounds, then Gary’s truck can haul $11,600 - 8,800 = 2,800$ pounds of gravel. <b>Answer: 2,800.</b>