Hyperbolic Fine-Tuning for Large Language Models

Menglin Yang[1,2], **Ram Samarth B B**[3], **Aosong Feng**[4], **Bo Xiong**[5]
Jiahong Liu[6], **Irwin King**[6], **Rex Ying**[4]
[1]HKUST(GZ); [2]HKUST; [3]Indian Institute of Science;
[4]Yale University; [5]Stanford University; [6]The Chinese University of Hong Kong
menglin.yang@outlook.com,ramsamarthbb@iisc.ac.in, aosong.feng@yale.edu,
xiongbo@stanford.edu, {jhliu22, king}@cse.cuhk.edu.hk, rex.ying@yale.edu
Code: https://github.com/marlin-codes/HypLoRA
Project:* https://hyperboliclearning.github.io/work/hyplora

Abstract

Large language models (LLMs) have demonstrated remarkable performance across various tasks. However, it remains an open question whether the default Euclidean space is the most suitable choice for LLMs. In this study, we investigate the geometric characteristics of LLMs, focusing specifically on tokens and their embeddings. Our findings reveal that token frequency follows a power-law distribution, where high-frequency tokens (e.g., "the," "that") constitute the minority, while low-frequency tokens (e.g., "apple," "dog") constitute the majority. Furthermore, high-frequency tokens cluster near the origin, whereas low-frequency tokens are positioned farther away in the embedding space. Additionally, token embeddings exhibit hyperbolic characteristics, indicating a latent tree-like structure within the embedding space. Motivated by these observations, we propose **HypLoRA**, an efficient fine-tuning approach that operates in hyperbolic space to exploit these underlying hierarchical structures better. **HypLoRA** performs low-rank adaptation directly in hyperbolic space, thereby preserving hyperbolic modeling capabilities throughout the fine-tuning process. Extensive experiments across various base models and reasoning benchmarks, specifically arithmetic and commonsense reasoning tasks, demonstrate that HypLoRA substantially improves LLM performance.

1 Introduction

Large language models (LLMs) such as GPT-4 [1], LLaMA [2], Gemma [3], and Qwen [4] have demonstrated remarkable capabilities in understanding and generating human-like text [5, 6, 7]. Despite their impressive capabilities, these models often rely on Euclidean geometry for token representation, which may inadequately capture the inherently complex and hierarchical nature of real-world data structures [8, 9, 10, 11, 12, 13]. Consider how words naturally organize into nested categories with varying levels of abstraction: abstract concepts like "fruit" occupy higher positions in the semantic hierarchy, while specific instances such as "apple" or "banana" populate 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 [11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23], offer promising alternatives for modeling hierarchical data. Hyperbolic space, distinguished by its negative curvature, is especially well-suited for representing tree-like hierarchical data due to its exponential volume growth and geometric prior. This geometric property makes hyperbolic space particularly capable for tasks involving complex, hierarchically structured information.

*Corresponding author: Menglin Yang.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

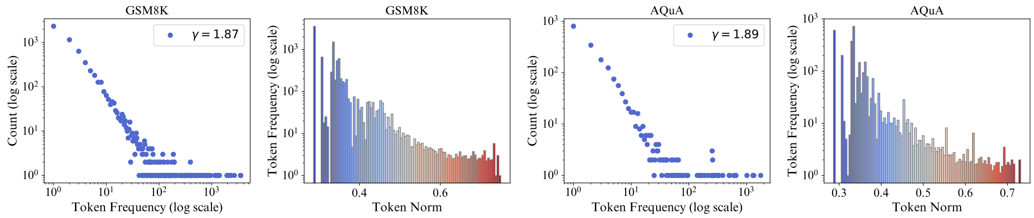


Figure 1: Token frequency distribution and token frequency vs. norm analysis for GSM8K (Group 1) and AQuA (Group 2) datasets in LLaMA3-8B. For each group, the left panels show the token frequency distributions (power-law distribution), while the right panels illustrate the relationship between token frequency and the corresponding norms. This visualization reveals the underlying geometric structure of the token embeddings. For additional data analysis and visualizations, please refer to Appendix A.

Proposed Analysis Framework. In this work, we first delve into how LLMs interact with token embeddings and explore the extent to which 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 and investigate how these frequencies are arranged across the embedding space. At the `local` level, we measure the hyperbolicity of the metric space spanned by each input prompt, where the hyperbolicity serves as a proxy for evaluating the distance or dissimilarity between the underlying embedding structure and a tree-like hierarchy [24, 25, 16].

Our analysis in Section 4 reveals several key insights. `Globally`, token frequency follows a power-law distribution, where high-frequency tokens (e.g., "the," "that") constitute the minority, while low-frequency tokens (e.g., "apple," "dog") constitute the majority. Power-law distributions are consistent with, and can naturally arise from, underlying hierarchical or branching generative mechanisms [12, 26, 27].[2] Besides, high-frequency tokens (e.g., abstract concepts, function words) tend to be located near the origin of the embedding space, while low-frequency tokens (e.g., specific terms) are farther away, as demonstrated in Table 1. `Locally`, our investigation of hyperbolicity (δ values) in Table 2 demonstrates that LLM token embeddings in each prompt exhibit significant tree-like properties.

Based on our findings above, a natural consideration is to develop hyperbolic LLMs that explicitly incorporate a hyperbolic inductive bias[3]. However, training LLMs from scratch is resource-intensive [29, 30, 31]. As a more resource-efficient alternative, we propose to build the first low-rank adaptation fine-tuning 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. By employing hyperbolic adapters on Euclidean LLMs for specific tasks, 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 within tangent space [32, 33, 34, 35, 36] 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[4]. 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.

Proposed Method. To address this limitation, we introduce **HypLoRA** to perform low-rank adaptation directly on the hyperbolic manifold without transformation to the tangent space, thus preserving hyperbolic modeling capabilities and counteracting the cancellation effect. HypLoRA integrates hyperbolic geometry into existing LLMs, implicitly introducing high-order interactions and accounting

[2]Power-law scaling alone does not uniquely identify the underlying structure or mechanism; additional evidence is needed to support a hierarchical interpretation [28].

[3]The connection between power-law distribution and hyperbolic geometry is elaborated in Section 4.3.

[4]The cancellation effect occurs because standard hyperbolic neural network [33, 32] approaches apply transformations in the tangent space at the origin, requiring the sequence: Euclidean embedding $\rightarrow$ exponential map to hyperbolic space $\rightarrow$ logarithmic map to tangent space $\rightarrow$ linear transformation $\rightarrow$ exponential map back to hyperbolic space $\rightarrow$ projection to Euclidean space. When these operations are chained together, the maps are mutually inverse and effectively cancel out, reducing the entire sequence to approximately the original Euclidean transformation BAx without preserving the beneficial hyperbolic geometry.

for token hierarchies, enabling them to benefit from hyperbolic characteristics while minimizing additional computational costs.

To summarize, our main contributions are twofold: (1) We conduct a comprehensive investigation into the geometric 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 keeping it aligned with the Euclidean LLM framework. We conduct extensive experiments on various models and different tasks, specifically arithmetic reasoning and commonsense reasoning, demonstrating clear advantages over competitive baselines.

2 Related Work

Hyperbolic Representation Learning and Foundation Models. Hyperbolic geometry has been successfully applied to various neural network architectures and models [19, 21, 18], including shallow hyperbolic neural networks [15, 33, 37, 38, 39, 40], hyperbolic CNNs [41, 42, 43], hyperbolic GNNs [32, 44, 45, 46], and hyperbolic attention networks or Transformers [47, 37, 38, 48]. These models leverage the inductive biases of hyperbolic geometry to achieve remarkable performance on various tasks and applications [32, 22, 23, 49, 16, 17, 50, 51, 52, 53]. Recent efforts have focused on adapting LLMs and CLIP [54] to hyperbolic spaces. Key advancements include developing more expressive hyperbolic image-text representations [55], enabling compositional entailment learning for deeper vision-language understanding [56], designing safety-aware hyperbolic frameworks for content moderation [57], and creating core modules to facilitate the construction of novel hyperbolic foundation models [58]. While these adaptations show promise, training LLMs from scratch remains computationally expensive [59, 60]. The computational complexity increases further when considering Riemannian optimization [59, 60, 61] and additional hyperbolic operations, like Möbius addition.

Geometric Analysis of Language Model Embeddings. Prior work has made important observations about the geometry of embeddings that helped shape and motivate our research. Reif et al. [62] demonstrated that BERT embeddings contain distinct syntactic and semantic subspaces and showed evidence of tree-like parse structures, while Gao et al. [63] revealed that token embeddings tend to cluster in a narrow cone during training, leading to representation degeneration. Building on these geometric insights, Rudman et al. [64] introduced IsoScore to formally quantify how uniformly embeddings utilize the ambient vector space. Additionally, Puccetti et al. [65] analyzed outlier dimensions in Transformers and showed their correlation with token frequencies. While these works provide crucial foundations for understanding embedding geometry, our work differs in that we specifically quantify and leverage the natural hyperbolicity of token embeddings.

Parameter-Efficient Fine-Tuning (PEFT) and LoRA. Fine-tuning LLMs [66, 1, 2] 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 comparable or even better performance compared to full fine-tuning. PEFT methods can be broadly categorized into prompt-based methods [67, 68, 69], adapter-based methods [70, 71], and reparameterization-based methods [31, 72, 73]. Among these, the reparameterization-based LoRA [31] has gained significant attention due to its simplicity, effectiveness, and compatibility with existing model architectures. Variants of LoRA, such as LoRA+[74], DoRA [75], and AdaLoRA [76], have been proposed to improve its performance and efficiency. Recent research has also investigated ensembles of multiple LoRAs [77, 78] and quantization techniques [79, 80, 81]. The proposed method is a foundational algorithm that is orthogonal to existing approaches and can potentially be combined with various LoRA variants to exploit their complementary strengths and achieve superior performance.

3 Preliminary

This section introduces the key concepts used in our study, including the Lorentz model of hyperbolic geometry and the LoRA adapter.

Hyperbolic Geometry. Unlike flat Euclidean geometry, hyperbolic geometry is characterized by a constant negative curvature. We utilize the Lorentz model, also known as the hyperboloid model due to its ability to effectively capture hierarchical structures and maintain numerical stability [14, 37, 82]. The Lorentz model in n dimensions with curvature $-1/K(K > 0)$ is defined as:

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

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 $\mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n$. It is defined as the set of all vectors $\mathbf{u}$ that are orthogonal to $\mathbf{x}$ under the Lorentzian inner product:

$$\mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n := \{\mathbf{u} \in \mathbb{R}^{n+1} : \langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}} = 0\}. \tag{2}$$

To facilitate projection between the hyperboloid and its tangent spaces at $\mathbf{x}$, one can 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 $\mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n$ 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 detailed formulas are given in Appendix C.

LoRA Adapter. The LoRA adapter provides an efficient approach for modifying 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 $\mathbf{x} \in \mathbb{R}^d$ into an output $\mathbf{z} \in \mathbb{R}^k$. 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 $\mathbb{R}^{k \times d}$, is adapted through a low-rank approximation. This is achieved by introducing an additional term, ΔW, to the original weight matrix:

$$\mathbf{z} = W_{\text{LoRA}}(\mathbf{x}) = W\mathbf{x} + \Delta W\mathbf{x} = W\mathbf{x} + BA\mathbf{x}. \tag{3}$$

Here, $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{k \times r}$ represent two smaller, learnable matrices where r is the rank of these matrices, which 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 $d \cdot k$ to $(d + k) \cdot r$, 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.

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 [83], AQuA [84], MAWPS [85], and SVAMP [86]. We also provide a broader analysis across different types of datasets and LLMs in Appendix A. Figure 1 (left) presents the distribution of token frequencies, with a power-law exponent $\gamma \approx 1.9$, as estimated by the `powerlaw` package [87]. In such distributions, the exponent γ controls how quickly token frequencies decline: smaller values of γ (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 is consistent with the tree-like hierarchical nature of language [11, 16, 88, 52, 23]. High-frequency tokens often correspond to more abstract or general concepts, while low-frequency tokens represent specific or rare terms. This pattern aligns with a hierarchical organization of the token space: abstract, high-frequency tokens cluster near the origin, while specific terms are positioned farther out, mirroring how general concepts sit at the core of a semantic hierarchy with specialized terms at the periphery.

Table 1: Mean, Minimum, and Maximum frequency and norm values of token embedding in different base models and groups. Group 1: *to, in, have, that, and, is, for*, Group 2: *how, much, many, time, cost*, Group 3: *animal, fruit, number, color, size*, Group 4: *dog, cow, apple, banana, 380, 480, purple, red, medium, small, large*.

Model	Group	Frequency (Mean [Min∼Max])	Norm (Mean [Min∼Max])
Gemma-7B	Group 1	4934.4 [1838 ∼ 8539]	3.160 [3.060 ∼ 3.299]
	Group 2	2709.4 [474 ∼ 6681]	3.561 [3.488 ∼ 3.627]
	Group 3	292.0 [34 ∼ 1191]	3.765 [3.623 ∼ 3.887]
	Group 4	114.3 [25 ∼ 284]	3.998 [3.660 ∼ 4.520]
LLaMA-7B	Group 1	4993.9 [1838 ∼ 8547]	0.951 [0.793 ∼ 1.060]
	Group 2	2712.6 [474 ∼ 6683]	1.222 [1.118 ∼ 1.299]
	Group 3	299.8 [34 ∼ 1200]	1.325 [1.274 ∼ 1.428]
	Group 4	139.1 [26 ∼ 286]	1.364 [1.326 ∼ 1.417]
LLaMA3-8B	Group 1	4937.4 [1838 ∼ 8547]	0.353 [0.330 ∼ 0.396]
	Group 2	2710.0 [474 ∼ 6683]	0.456 [0.394 ∼ 0.499]
	Group 3	292.6 [34 ∼ 1191]	0.499 [0.452 ∼ 0.549]
	Group 4	97.1 [13 ∼ 284]	0.569 [0.499 ∼ 0.675]
LLaMA-13B	Group 1	4993.9 [1838 ∼ 8547]	1.027 [0.833 ∼ 1.255]
	Group 2	2712.6 [474 ∼ 6683]	1.429 [1.346 ∼ 1.489]
	Group 3	299.8 [34 ∼ 1200]	1.494 [1.453 ∼ 1.532]
	Group 4	139.1 [26 ∼ 286]	1.501 [1.470 ∼ 1.526]

Empirical Observation. 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. As shown in Figure 1 (right), high-frequency tokens (e.g., “the,” “that”) tend to have smaller norms, while low-frequency tokens (e.g., “apple,” “dog”) have larger norms. Table 1 presents representative tokens across different frequencies and norm ranges within the embedding space of different base models. We categorize tokens into four groups based on their linguistic function and specificity: Group 1 contains high-frequency function words (e.g., *to, is, and*), Group 2 contains common question/quantity words (e.g., *how, much, many*), Group 3 contains general category nouns (e.g., *animal, fruit, color*), and Group 4 contains specific instances (e.g., *dog, apple, purple*).

The results presented in Table 1 demonstrate several critical findings. *First*, we observe a statistically significant separation between functional/abstract words (Group 1) and specific terms (Group 4) across all models, with Group 1 consistently exhibiting the smallest embedding norms and highest frequencies, while Group 4 shows the largest norms and lowest frequencies. *Second*, the relative ordering of groups remains consistent across all examined models, with Group 1 $<$ Group 2 $<$ Group 3 $<$ Group 4 in terms of embedding norms, despite absolute magnitude variations. Most notably, even across different architectural families (LLaMA vs. Gemma), the hierarchical organization principle remains preserved, though with different absolute scales, where Gemma-7B exhibits systematically larger embedding norms (mean Group 1 norm: 3.160) compared to LLaMA models (mean Group 1 norm: $0.353 \sim 1.027$), yet maintains the same relative hierarchical structure.

Conclusion (1) These findings suggest that the spatial organization of token embeddings reflects the inherent hierarchical relationships in language, supporting the hypothesis that token embedding in LLMs exhibits a tree-like structure, with spatial positioning aligned with token frequency and specificity. It is worth noting, however, that a power-law distribution of token frequency alone does not guarantee the emergence of a hierarchical token embedding, as it also depends on the training objectives. Our analysis demonstrates that the hierarchy is strongly correlated with token frequencies, which can be understood through the lens of LLMs' tokenization and co-occurrence pattern learning during training [89]. While the exact mechanisms underlying this relationship require further investigation in future work, the spatial distribution of token embeddings remains crucial as it provides the primary motivation for our methodological approach.

4.2 δ-Hyperbolicity of Local Token Embeddings

To rigorously quantify the hierarchical nature of token embeddings, we further examine the δ-hyperbolicity of the space spanned by the token embedding. δ-hyperbolicity, introduced by Gromov [90], is a measure that captures the degree to which a metric space deviates from an exact tree

Table 2: Comparison of δ-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(δ)**
Sphere Space	0.99 ± 0.01
Random Graph	0.62 ± 0.34
PubMed Graph	0.40 ± 0.45
Scale-free Graph	0.00
Tree Graph	0.00

Hyperbolicity(δ)	MAWPS	SVAMP	GSM8K	AQuA
LLaMA-7B	0.08 ± 0.02	0.09 ± 0.01	0.10 ± 0.01	0.10 ± 0.01
LLaMA-13B	0.08 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.10 ± 0.01
Gemma-7B	0.11 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.12 ± 0.01
LLaMA3-8B	0.06 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.08 ± 0.01
Average	0.08 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.10 ± 0.01

structure. Lower values of δ imply a space more similar to a perfect tree, while higher values indicate deviation from a tree-like structure.

We compute δ-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:

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

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

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

Quantitative Analysis. To measure the hyperbolicity of token embeddings, we apply this algorithm to various open-source LLMs. Following the methodologies proposed by Khrulkov et al. [16] and Cetin et al. [17], we estimate δ-hyperbolicity using the efficient algorithm introduced by Fournier et al. [91]. To ensure scale invariance, we normalize δ by the diameter of the embedding space, $\mathrm{diam}(X)$, yielding a relative measure: $\delta_{rel} = \frac{2\delta}{\mathrm{diam}(X)}$. 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. We employ Euclidean distance as a measure of the shortest distance, maintaining the same computational paradigm as in previous works [16, 17]. To further validate the correctness of this approach, we generate a series of random graphs with predefined hyperbolicity, embed them using a two-layer graph neural network (GNN) [92], and then compute the hyperbolicity. 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.

Conclusion (2) 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. We also provide the hyperbolicity analysis of the final hidden layer in Appendix A.3.

4.3 Connection between Power-law Distribution and Hyperbolic Geometry

Having established both the global power-law distribution (Section 4.1) and local tree-like geometry (Section 4.2) of token embeddings, we now examine the theoretical connection between these two observations.

The observation of a power-law distribution in token frequencies, as discussed in Section 4, is not merely a statistical curiosity. It has deep connections to the underlying geometry of the data, particularly to hyperbolic spaces, which are well-suited for representing hierarchical structures [11, 12, 93, 94]. For instance, Nickel and Kiela [11] highlighted that the existence of power-law degree distributions can often be traced back to hierarchical structures. Similarly, Ravasz and Barabási [88] established that the scaling law $P(k) \sim k^{-\gamma}$ can signify the co-existence of a hierarchy of nodes with varying degrees of clustering. Krioukov et al. [12] further strengthened this connection by showing that the exponent of the power-law degree distribution is a function of the hyperbolic space curvature. Building on this geometric understanding, Papadopoulos et al. [94] demonstrated that complex (scale-free) network topologies naturally emerge when networks grow within an underlying hyperbolic metric space, and importantly, that the resulting hyperbolic embedding of these dynamic scale-free networks facilitates highly efficient greedy forwarding.

To formalize this connection with hyperbolic geometry, we can consider embedding tokens in a hyperbolic space. A common model for hyperbolic space is the Poincaré disk model ($\mathbb{H}^2$) with curvature $K = -1$[5]. In such a space, both the circumference $C(r)$ and area $A(r)$ of a circle of radius r exhibit exponential growth:

$$C(r) = 2\pi \sinh(r) \sim e^r \quad \text{as } r \to \infty, \tag{6}$$

$$A(r) = 2\pi(\cosh(r) - 1) \sim e^r \quad \text{as } r \to \infty. \tag{7}$$

If we consider token embeddings in a hyperbolic space with polar coordinates (r, θ), where $r \in \mathbb{R}^+$ is the radial coordinate (correlating with token frequency) and $\theta \in [0, 2\pi]$ is the angular coordinate (encoding semantic similarity), the radial distribution of tokens follows $p(r) \sim e^{-\zeta r}$, where $\zeta > 0$ relates to the hyperbolic curvature K. The frequency function $k(r)$ for tokens at radius r is then given by $k(r) \sim e^{-r}$. Given $k(r) \sim e^{-r}$, we have $r \sim -\ln k$, and thus $\left|\frac{dr}{dk}\right| \sim k^{-1}$. Combined with the radial distribution $p(r) \sim e^{-\zeta r} \sim k^{\zeta}$, this yields:

$$P(k) \sim p(r) \left|\frac{dr}{dk}\right| \sim k^{\zeta} \cdot k^{-1} \sim k^{-(1-\zeta)}. \tag{8}$$

Following the parameterization of Krioukov et al. [12], the power-law exponent γ relates to the curvature parameter via $\gamma = 2/\zeta + 1$ in the context of complex networks. This relationship underscores the theoretical connection between the power-law behavior observed in token frequencies and the inherent hyperbolic geometry of the embedding space. Since hyperbolic models such as the Poincaré ball model and the Lorentz model are isometric, this conclusion can be extended to other hyperbolic models.

Hyperbolic space offers distinct advantages for modeling language hierarchies, especially when addressing the structural and spatial constraints of token co-occurrence: **(1) Separation of Low-Frequency Tokens.** Tokens with low frequencies, which typically represent more specific or granular concepts, require clear separation from each other to maintain semantic clarity. **(2) Proximity to High-Frequency Hypernyms.** Simultaneously, these low-frequency tokens should remain close to their corresponding high-frequency hypernyms or function words. Hyperbolic space is uniquely suited for capturing these dual constraints due to its exponential volume growth, which inherently supports hierarchical structure and allows for ample separation of specific entities while keeping them close to their parent categories. This contrasts with Euclidean space, where such arrangements can lead to crowding or distortion of distances.

Overall Conclusion. Through these analyses, we demonstrate that token embeddings in LLMs exhibit hierarchical organization and significant hyperbolicity. This understanding not only sheds light on the geometric nature of token embeddings but also motivates the development of methods that can better capture and preserve these underlying geometric properties.[6]

5 Hyperbolic Fine-Tuning for LLMs

The core technique in the LoRA adapter involves matrix transformations. The conventional approach to implementing these transformations in the Lorentz model of hyperbolic geometry is through operations in the tangent space, while maintaining the learnable weights in Euclidean space [33, 32].

[5]The derivation uses the Poincaré ball model for its intuitive geometric interpretation. However, all models of hyperbolic space, including the Poincaré ball, the Lorentz (hyperboloid) model, and the Klein model, are isometrically equivalent, preserving geodesic distances under explicit diffeomorphisms [95, 96]. Since our method (Section 5) operates in the Lorentz model, the theoretical connections established here between power-law distributions and hyperbolic geometry remain fully applicable.

[6]While our analysis reveals consistent hierarchical patterns across multiple LLMs, several limitations should be noted. First, our investigation focuses on arithmetic reasoning and commonsense datasets (please check Appendix A for details); the generalizability to other domains (e.g., code, multilingual text) requires further validation. Second, the relationship between token frequency and embedding norm, while strong, is correlational rather than causal. Third, our δ-hyperbolicity measurements are computed at the prompt level; corpus-level analysis may yield different insights.

Table 3: Comparison of various LLMs on arithmetic reasoning tasks. 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 (since the datasets are imbalanced). For more adapter comparisons, please see Appendix F.

Base Model	PEFT Method	# Params (%)	MAWPS(8.5%)	SVAMP(35.6%)	GSM8K(46.9%)	AQuA(9.0%)	M.AVG
GPT-3.5	None	None	87.4	69.9	56.4	38.9	62.3
LLaMA-7B	LoRA	0.83	**81.9**	48.2	38.3	18.5	43.7
	HypLoRA (Ours)	0.83	79.0	**49.1**	**39.1**	**20.5**	**44.4**
LLaMA-13B	LoRA	0.67	**83.5**	54.7	48.5	18.5	51.0
	HypLoRA (Ours)	0.67	83.2	**54.8**	**49.0**	**21.5**	**51.5**
Gemma-7B	LoRA	0.79	**91.6**	76.2	66.3	28.9	68.6
	HypLoRA (Ours)	0.79	89.5	**78.7**	**69.5**	**32.7**	**71.2**
LLaMA3-8B	LoRA	0.70	**92.7**	78.9	70.8	30.4	71.9
	HypLoRA (Ours)	0.70	91.6	**80.5**	**74.0**	**34.2**	**74.2**
Gemma3-4B	LoRA	1.04	**90.8**	77.3	72.3	50.8	73.7
	HypLoRA (Ours)	1.04	88.2	**83.9**	**76.1**	**53.2**	**77.8**
Qwen2.5-7B	LoRA	0.71	90.8	84.4	78.6	68.1	80.8
	HypLoRA (Ours)	0.71	**91.2**	**92.2**	**87.9**	**71.6**	**88.3**

However, this approach presents a significant challenge for our application. Since the hidden states of LLMs exist in Euclidean space, we would need to project these states to hyperbolic space and subsequently map them back to the tangent space. This process results in consecutive logarithmic and exponential mappings $(\log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\mathbf{x})))$, which effectively cancel each other out, reducing the method to the original LoRA approach and nullifying any benefits from hyperbolic geometry.

Direct Lorentz Low-Rank Transformation (LLR). To overcome this limitation, we propose a direct Lorentz Low-Rank Transformation (LLR) that operates directly on the hyperbolic space without relying on tangent space mappings. This approach allows us to perform low-rank adaptation while preserving the advantages of hyperbolic geometry:

$$\begin{aligned} \mathbf{z}^E &= W_{\text{LoRA}}(\mathbf{x}^E) = W\mathbf{x}^E + \Delta W\mathbf{x}^E \\ &= W\mathbf{x}^E + \Pi_{\log}^K(\mathbf{LLR}(BA, \Pi_{\exp}^K(\mathbf{x}^E))), \end{aligned} \tag{9}$$

where $\mathbf{LLR}$ represents the direct Lorentz Low-Rank Transformation that operates directly on the hyperbolic representation $\mathbf{x}^H = \Pi_{\exp}^K(\mathbf{x}^E)$:

$$\mathbf{LLR}(BA, \mathbf{x}^H) = (\sqrt{\|BA\mathbf{x}_s^H\|_2^2 + K}, BA\mathbf{x}_s^H), \tag{10}$$

where $\mathbf{x}_s^H$ is the space-like component of $\mathbf{x}^H$, i.e., $\mathbf{x}_s^H = \mathbf{x}_{[1:n]}^H$ without the first time-like dimension $\mathbf{x}_{[0:1]}^H$. The operators $\Pi_{\exp}^K$ and $\Pi_{\log}^K$ represent projections from a local tangent space to hyperbolic space (e.g., exponential map) and from hyperbolic space to a local tangent space (e.g., logarithmic map), respectively. The detailed formulas are provided in Appendix C. It can be verified that $\mathbf{LLR}(BA, \mathbf{x}^H) \in \mathcal{L}^n$, ensuring that our transformation remains within the Lorentz model of hyperbolic space. This transformation primarily affects the space-like dimensions, functioning similarly to a pseudo-Lorentz rotation [37]. The linear transformation is inspired by hyperbolic neural networks [37, 48, 97]. For efficient integration with LLMs, the transformation removes normalization and non-linear activation terms in [37], varying curvatures in [48], and orthogonal constraints in [97]. Our main contribution lies in applying hyperbolic low-rank adaptation for LLMs, while the specific linear transformation itself is flexible, and other transformations on the manifold could also be compatible with our approach.

By adapting in the hyperbolic domain, HypLoRA captures more complex hierarchical relationships than traditional Euclidean-based methods, as detailed in Proposition 1. Additionally, the low-rank nature of the adaptation matrices A and B promotes parameter efficiency, making HypLoRA well-suited for LLMs.

Time Complexity Analysis. 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 space mapping. These additional operations, nevertheless, can be completed within $\mathcal{O}(N)$ where N is the number of input tokens.

Table 4: Comparison of various LLMs on commonsense reasoning tasks. These datasets contain relatively similar amounts of data, so we use AVG to represent the average accuracy.

Base Model	PEFT Method	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	AVG
GPT-3.5	None	None	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA3-8B	LoRA	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	HypLoRA (Ours)	0.70	**74.1**	**87.6**	**80.6**	**94.5**	**84.7**	**90.4**	**81.2**	**85.2**	**84.8**
Gemma3-4B	LoRA	1.04	68.1	83.2	77.2	88.9	**80.5**	84.5	69.9	83.6	79.5
	HypLoRA (Ours)	1.04	**70.0**	**84.3**	**79.2**	**91.5**	80.3	**89.1**	**75.9**	**86.4**	**82.5**
Qwen2.5-7B	LoRA	0.71	**73.4**	**89.5**	79.5	93.6	84.1	92.8	82.0	87.0	85.2
	HypLoRA (Ours)	0.71	72.8	89.3	**79.8**	**94.8**	**84.4**	**95.5**	**87.5**	**90.8**	**87.0**

Proposition 1. *Let $\mathbf{x} \in \mathbb{R}^d$ denote the input token embeddings. The HypLoRA adaptation, applied to $\mathbf{x}$, involves a sequence of projection into hyperbolic space, a Direct Lorentz Low-Rank Transformation (LLR), and projection back to Euclidean space. Due to the non-linear nature of these hyperbolic operations, the effective transformation applied by HypLoRA introduces higher-order terms with respect to $\mathbf{x}$. As detailed in Appendix E, these terms exhibit explicit dependency on the L2 norm, $\|\mathbf{x}\|_2$, of the input embeddings. This norm-dependent, higher-order modification enables HypLoRA to capture hierarchical relationships in the embedding space, thereby achieving natural alignment with the underlying hyperbolic geometry of the token representations.*

5.1 Experimental Settings

Datasets. Following the experiment setup outlined in [98], we utilize two high-quality datasets, Math10K and Commonsense170K, tailored for mathematical and commonsense reasoning, respectively. Math10K consists of training data from GSM8K [83], MAWPS, MAWPS-single [85], and 1,000 samples from AQuA [84], augmented with ChatGPT-generated step-by-step rationales to reinforce reasoning capabilities. The test set includes GSM8K, AQuA, MAWPS, and SVAMP [86], ensuring no overlap with the training data. Commonsense170K is constructed by reformatting samples from BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-e, ARC-c, and OBQA using standardized templates that outline the task, content, and answer, resulting in 170K training samples. The test datasets are drawn from the same sources, with strict separation from training samples. For fine-tuning methods, we compare with LoRA [31] and also make a comparison with other adapters in Appendix F, which also includes training details.

5.2 Experimental Results

Table 3 summarizes our key experimental outcomes on arithmetic reasoning tasks, while Table 4 presents results for commonsense reasoning benchmarks. Our primary comparison contrasts LoRA and HypLoRA to demonstrate the effectiveness of the proposed approach, with additional baselines provided in Appendix F.

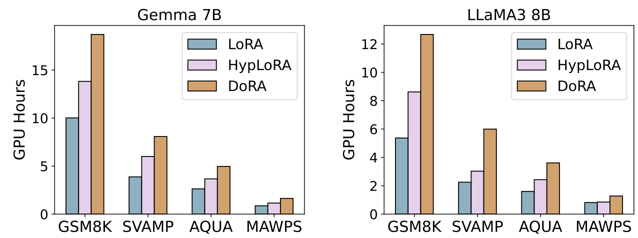
Arithmetic Reasoning Performance. On arithmetic reasoning tasks, as indicated by results in Table 3, HypLoRA shows notable efficacy, especially on datasets recognized for their complexity, such as GSM8K, AQuA, and SVAMP. These datasets demand robust multi-step reasoning and a nuanced understanding of numerical and textual relationships. For instance, on Qwen2.5-7B, HypLoRA achieves a substantial $+7.5$ percentage point improvement in M.AVG (88.3% vs. 80.8%), with notable gains of $+7.8\%$ on SVAMP and $+9.3\%$ on GSM8K. The enhanced performance of HypLoRA in these areas aligns with its design; by operating in hyperbolic space, it can better model the hierarchical structure of problems and distinguish subtle yet critical differences in input embeddings. This is further corroborated by the theoretical analysis (Appendix E), which posits that HypLoRA introduces higher-order, norm-dependent terms. These terms allow the model to develop a more refined sensitivity to token importance and inter-token relationships.

Commonsense Reasoning Performance. The robust performance of HypLoRA extends to commonsense reasoning, as detailed in Table 4. For the Gemma3-4B model, HypLoRA achieved an average accuracy of 82.5% across all datasets, surpassing LoRA's 79.5%. Similarly, on the Qwen2.5-7B model, HypLoRA obtained an average of 87.0% compared to LoRA's 85.2%. These improvements are distributed across various commonsense benchmarks, including notable gains on datasets like ARC-c and OBQA for Gemma3-4B, and ARC-c, ARC-e, and OBQA for Qwen2.5-7B. Commonsense reasoning often relies on understanding implicit relationships and contextual nuances, which may not

Table 5: Results for varying curvature K on the Gemma3-4B model

Dataset	K=0.5	K=1.0
MAWPS	88.2	**91.9**
SVAMP	**83.9**	80.3
GSM8K	**76.1**	73.8
AQuA	**53.5**	52.7
M.AVG	**77.8**	75.8

Figure 2: GPU (A100) usage during inference



always be explicitly hierarchical but still benefit from the richer representational capacity offered by hyperbolic geometry. The ability of HypLoRA to better discern these subtleties, likely due to the mechanisms described in Proposition 1, contributes to these observed performance gains, showcasing the broad applicability of hyperbolic fine-tuning.

The Impact of Curvature on Performance. Curvature in hyperbolic space is a key hyperparameter in HypLoRA, directly affecting its capacity to model underlying structures and geometries. To evaluate its impact, we experiment with a learnable curvature initialized with different curvature values on the Gemma3-4B model, as shown in Table 5, where the curvature is defined as $-1/K$. Our results demonstrate that curvature does influence model performance. For Gemma-7B and Gemma3-4B, a curvature value of 0.5 consistently yields the best overall performance across both arithmetic and commonsense reasoning benchmarks. Similarly, for LLaMA3-8B, 0.5 proves optimal. In commonsense reasoning benchmarks, a curvature of 1.0 performs best for LLaMA3-8B and Qwen2.5-7B.

Efficiency. In Section 5, we analyze the time complexity of our approach, which remains consistent with that of LoRA. However, during actual inference, HypLoRA incurs additional computational overhead due to operations such as projections. These operations introduce some additional runtime, particularly for larger models. The GPU hours for inference on four datasets are presented in Figure 2. Despite this overhead, our method demonstrates improved efficiency when compared to the previous competitive model, DoRA. Notably, HypLoRA still outperforms DoRA in terms of both runtime and overall efficiency. Besides, all models can be fine-tuned in approximately one hour for optimal training efficiency.

6 Conclusion

In this work, we investigated the non-Euclidean geometric properties inherent in LLMs, confirming their strong hyperbolic characteristics, which suggest underlying hierarchical structures. Building on these insights, we introduced HypLoRA, a hyperbolic low-rank adaptation technique. HypLoRA performs fine-tuning directly on the hyperbolic manifold. Extensive experiments show that HypLoRA significantly improves LLM performance on arithmetic reasoning and commonsense tasks. 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.

Broader Impact. Enhancing reasoning-oriented LLMs can help education, scientific assistance, and safer decision-support systems, but the same improvements may also accelerate misuse (e.g., automating complex disinformation or amplifying biased advice) and increase energy consumption due to added hyperbolic projections. We therefore advocate releasing checkpoints and code with usage guidelines (as in our public repo), tracking compute budgets when scaling HypLoRA further.

Acknowledgment

The authors would like to express their sincere gratitude to the anonymous reviewers and the Area Chair for their valuable comments and insightful suggestions, which have greatly improved this work. We thank our colleagues and lab mates at HKUST(GZ), IISc, Yale, Stanford, and CUHK for motivating discussions on geometric learning for LLMs, and we acknowledge the administrators of the institutional GPU clusters that enabled the large-scale experiments. Part of this research was conducted while Menglin Yang was at Yale University, whose support and computing resources were instrumental in shaping the initial ideas of this project.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] 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.

[3] Google Deepmind Gemma Team. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[4] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[5] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

[6] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

[7] Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*, 2025.

[8] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[9] Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *International Conference on Machine Learning (ICML)*, pages 486–496. PMLR, 2020.

[10] Atsushi Suzuki, Atsushi Nitanda, Jing Wang, Linchuan Xu, Kenji Yamanishi, and Marc Cavazza. Generalization error bound for hyperbolic ordinal embedding. In *International Conference on Machine Learning (ICML)*, pages 10011–10021. PMLR, 2021.

[11] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 6338–6347, 2017.

[12] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82(3):036106, 2010.

[13] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.

[14] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning (ICML)*, pages 3779–3788, 2018.

[15] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning (ICML)*, pages 1646–1655. PMLR, 2018.

[16] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6418–6428, 2020.

[17] Edoardo Cetin, Benjamin Chamberlain, Michael Bronstein, and Jonathan J Hunt. Hyperbolic deep reinforcement learning. *arXiv preprint arXiv:2210.01542*, 2022.

[18] 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*, 2021.

[19] Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. Hyperbolic graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852*, 2022.

[20] Jiahong Liu, Xinyu Fu, Menglin Yang, Weixi Zhang, Rex Ying, and Irwin King. Client-specific hyperbolic federated learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2024.

[21] Pascal Mettes, Mina Ghadimi Atigh, Martin Keller-Ressel, Jeffrey Gu, and Serena Yeung. Hyperbolic deep learning in computer vision: A survey. *arXiv preprint arXiv:2305.06611*, 2023.

[22] Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. Hrcf: Enhancing collaborative filtering via hyperbolic geometric regularization. In *The Web Conference (WWW)*, pages 2462–2471, 2022.

[23] Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. Hicf: Hyperbolic informative collaborative filtering. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2212–2221, 2022.

[24] Michele Borassi, Alessandro Chessa, and Guido Caldarelli. Hyperbolicity measures democracy in real-world networks. *Physical Review E*, 92(3):032812, 2015.

[25] W Sean Kennedy, Onuttom Narayan, and Iraj Saniee. On the hyperbolicity of large-scale networks. *arXiv preprint arXiv:1307.0031*, 2013.

[26] Albert-László Barabási, Zoltán Dezső, Erzsébet Ravasz, Soon-Hyung Yook, and Zoltán Oltvai. Scale-free and hierarchical structures in complex networks. In *AIP Conference Proceedings*, volume 661, pages 1–16. American Institute of Physics, 2003.

[27] Enrique Alvarez-Lacalle, Beate Dorow, J-P Eckmann, and Elisha Moses. Hierarchical structures induce long-range dynamical correlations in written texts. *Proceedings of the National Academy of Sciences*, 103(21):7956–7961, 2006.

[28] Kai Nakaishi, Ryo Yoshida, Kohei Kajikawa, Koji Hukushima, and Yohei Oseki. Rethinking the relationship between the power law and hierarchical structures. *arXiv preprint arXiv:2505.04984*, 2025.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[30] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[31] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[32] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[33] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[34] Menglin Yang, Min Zhou, Hui Xiong, and Irwin King. Hyperbolic temporal network embedding. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2022.

[35] Sungjun Cho, Seunghyuk Cho, Sungwoo Park, Hankook Lee, Honglak Lee, and Moontae Lee. Curve your attention: Mixed-curvature transformers for graph representation learning. *arXiv preprint arXiv:2309.04082*, 2023.

[36] Xingcheng Fu, Yisen Gao, Yuecen Wei, Qingyun Sun, Hao Peng, Jianxin Li, and Xianxian Li. Hyperbolic geometric latent diffusion model for graph generation. *arXiv preprint arXiv:2405.03188*, 2024.

[37] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. *arXiv preprint arXiv:2105.14686*, 2021.

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

[39] Xiaomeng Fan, Yuwei Wu, Zhi Gao, Mehrtash Harandi, and Yunde Jia. Curvature learning for generalization of hyperbolic neural networks. *International Journal of Computer Vision (IJCV)*, pages 1–37, 2025.

[40] Yidan Mao, Jing Gu, Marcus C Werner, and Dongmian Zou. Klein model for hyperbolic neural networks. *arXiv preprint arXiv:2410.16813*, 2024.

[41] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Hyperbolic geometry in computer vision: A novel framework for convolutional neural networks. *arXiv preprint arXiv:2303.15919*, 2023.

[42] Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincaré resnet. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5419–5428, 2023.

[43] Raiyan R Khan, Philippe Chlenski, and Itsik Pe'er. Hyperbolic genome embeddings. *arXiv preprint arXiv:2507.21648*, 2025.

[44] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[45] Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Ye. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 8(6):1690–1701, 2021.

[46] Menglin Yang, Min Zhou, Lujia Pan, and Irwin King. κhgcn: Tree-likeness modeling via continuous and discrete curvature learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2965–2977, 2023.

[47] 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.

[48] Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. *arXiv preprint arXiv:2407.01290*, 2024.

[49] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. HGCF: Hyperbolic graph convolution networks for collaborative filtering. In *The Web Conference (WWW)*, pages 593–601, 2021.

[50] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2603–2612, 2021.

[51] Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. Hyperbolic embedding inference for structured multi-label prediction. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 33016–33028, 2022.

[52] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1975–1985, 2021.

[53] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Curvature generation in curved spaces for few-shot learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8691–8700, 2021.

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[55] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam. Hyperbolic image-text representations. In *International Conference on Machine Learning (ICML)*, 2023.

[56] Avik Pal, Max van Spengler, Guido Maria D'Amely di Melendugno, Alessandro Flaborea, Fabio Galasso, and Pascal Mettes. Compositional entailment learning for hyperbolic vision-language models. *arXiv preprint arXiv:2410.06912*, 2024.

[57] Tobia Poppi, Tejaswi Kasarla, Pascal Mettes, Lorenzo Baraldi, and Rita Cucchiara. Hyperbolic safety-aware vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[58] Neil He, Menglin Yang, and Rex Ying. Hypercore: The core framework for building hyperbolic foundation models with comprehensive modules. In *International Conference on Learning Representations (ICLR)*, 2025.

[59] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*, 2020.

[60] Steven Thomas Smith. Optimization techniques on riemannian manifolds. *arXiv preprint arXiv:1407.5965*, 2014.

[61] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

[62] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[63] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019.

[64] William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. Isoscore: Measuring the uniformity of embedding space utilization. *arXiv preprint arXiv:2108.07344*, 2021.

[65] Giovanni Puccetti, Anna Rogers, Aleksandr Drozd, and Felice Dell'Orletta. Outliers dimensions that disrupt transformers are driven by frequency. *arXiv preprint arXiv:2205.11380*, 2022.

[66] OpenAI Foundation. Introducing chatgpt. https://openai.com/index/chatgpt, November 2022.

[67] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[68] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[69] Yujia Qin, Xiaozhi Wang, Yusheng Su, Yankai Lin, Ning Ding, Jing Yi, Weize Chen, Zhiyuan Liu, Juanzi Li, Lei Hou, et al. Exploring universal intrinsic task subspace via prompt tuning. *arXiv preprint arXiv:2110.07867*, 2021.

[70] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, pages 2790–2799. PMLR, 2019.

[71] Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 2021.

[72] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[73] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. Krona: Parameter efficient tuning with kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022.

[74] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

[75] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

[76] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

[77] Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

[78] Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.17263*, 2024.

[79] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

[80] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

[81] Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023.

[82] Gal Mishne, Zhengchao Wan, Yusu Wang, and Sheng Yang. The numerical stability of hyperbolic representation learning. In *International Conference on Machine Learning (ICML)*, pages 24925–24949. PMLR, 2023.

[83] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[84] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

[85] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1152–1157, 2016.

[86] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

[87] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. Powerlaw: a python package for analysis of heavy-tailed distributions. *PloS one*, 9(1):e85777, 2014.

[88] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical review E*, 67(2):026112, 2003.

[89] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (ACL)*, pages 1715–1725, 2016.

[90] Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer, 1987.

[91] 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.

[92] TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[93] Dmitri Krioukov, Fragkiskos Papadopoulos, Amin Vahdat, and Marián Boguná. Curvature and temperature of complex networks. *Physical Review E*, 80(3):035101, 2009.

[94] Fragkiskos Papadopoulos, Dmitri Krioukov, Marián Boguná, and Amin Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *Infocom*, pages 1–9. IEEE, 2010.

[95] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

[96] John G Ratcliffe. *Foundations of hyperbolic manifolds*, volume 149. Springer, 2006.

[97] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 154–163, 2021.

[98] Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[99] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[100] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *SciPy2008*, pages 11–15. Pasadena, CA USA, 2008.

[101] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. *arXiv preprint arXiv:1911.08411*, 2019.

[102] Philip McCord Morse and Herman Feshbach. *Methods of theoretical physics*. Technology Press, 1946.

[103] Valter Moretti. The interplay of the polar decomposition theorem and the lorentz group. *arXiv preprint math-ph/0211047*, 2002.

[104] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The paper's abstract and introduction explicitly state the main claims, including the contribution of the proposed method and analysis, and its scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses a limitation in the Conclusion (Section 6).

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The paper presents Proposition 1 regarding HypLoRA, introducing higher-order terms. It states that the details and derivation of these terms are provided in Appendix F.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5.1 "Experimental Settings" describes the datasets, models, and implementation details, including optimizer (AdamW), etc.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The datasets used (e.g., GSM8K, AQUA, various commonsense reasoning benchmarks) are publicly available and cited.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5.1 details the experimental setup.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Table 2, which presents hyperbolicity across various metric spaces and datasets, includes error bars reported as mean and standard deviation. Tables 3 and 4 report mean accuracies over three runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides implementation details in Section 5.1, including the GPU type (NVIDIA A100) used for experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Based on the content of the paper, the research focuses on investigating geometric properties of LLMs and proposing a new fine-tuning method. The datasets used are standard benchmarks in the field. There is no indication of activities that would violate the NeurIPS Code of Ethics, such as plagiarism, falsification of data, or unethical use of human subjects.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Conclusion part.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: It does not introduce new large-scale pretrained models or novel scraped datasets that would pose a high risk for misuse requiring specific safeguards beyond those applicable to the original models it builds upon.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The creators/original owners of assets (datasets like GSM8K, MAWPS, AQUA; models like LLaMA, Gemma; and software like the Powerlaw Package) are credited via citations.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: No new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The datasets are existing benchmarks.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The research does not appear to involve human subjects in a way that would necessitate IRB approval, as per the justification for question 14.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [No]

    Justification: No, except for writing, editing, or formatting purposes.

    Guidelines:

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

Appendix Contents

A More Investigation Results

A.1 Token Frequency and Norm Distribution on Mathematical Reasoning

To provide a comprehensive understanding of the geometric properties of token embeddings across different mathematical reasoning tasks, we extend our analysis beyond the GSM8K dataset presented in the main text to include AQuA and MAWPS datasets. This broader investigation allows us to validate the consistency of our findings across diverse mathematical problem types and complexity levels. The AQuA dataset presents algebraic word problems that require multi-step reasoning and equation solving, while MAWPS focuses on elementary arithmetic word problems with varying structural complexity. By analyzing token distributions across these complementary datasets, we can assess whether the observed power-law behavior and hierarchical token organization represent universal properties of mathematical reasoning tasks or are specific to particular problem domains.

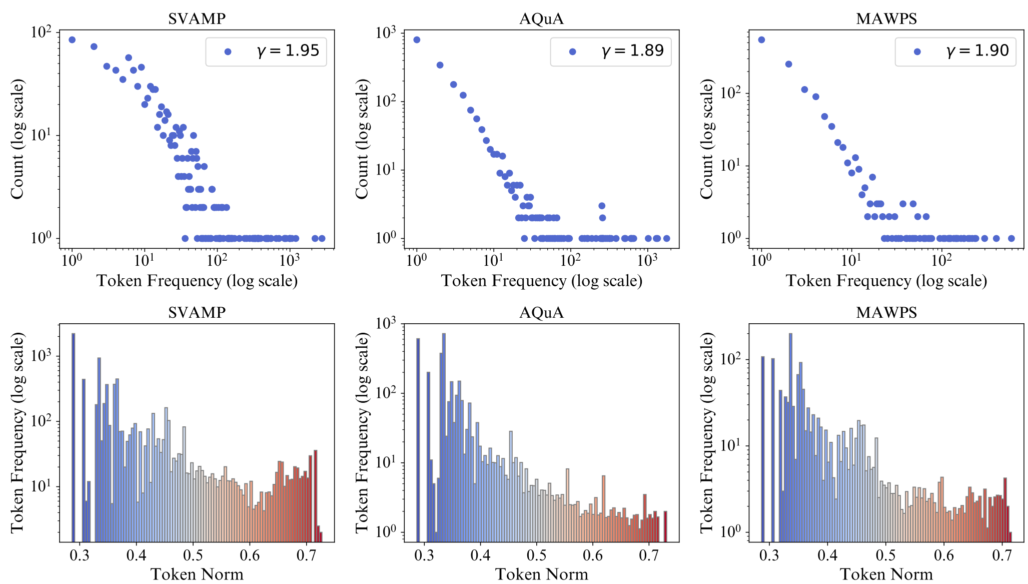


Figure 3: Token frequency distribution (top row) and token frequency vs. norm (bottom row) across different mathematical reasoning datasets in LLaMA3. The top row shows the power-law distribution of token frequencies with the decay rate (γ) annotated for each dataset. The bottom row illustrates the relationship between token frequency and token norm, binned and colored by frequency, where higher token norms correspond to lower frequencies.

Our extended analysis, illustrated in Figure 3, reveals remarkably consistent patterns across all three mathematical reasoning datasets. The power-law exponents remain stable within a narrow range ($\gamma \in [1.89, 1.95]$), indicating that the hierarchical structure of mathematical language is preserved regardless of the specific problem type or complexity level. The relationship between token frequency and embedding norms shows consistent inverse correlation across all datasets, with high-frequency mathematical operators and common function words clustering near the origin, while domain-specific mathematical terms and numerical values are positioned at greater distances. **This consistency strengthens our hypothesis that mathematical reasoning tasks inherently exhibit hyperbolic characteristics in their token embedding spaces**, providing strong empirical support for the effectiveness of hyperbolic fine-tuning approaches like HypLoRA in mathematical domains.

A.2 Token Frequency and Norm Distribution on Commonsense Reasoning

To demonstrate the generalizability of our findings beyond mathematical reasoning, we conduct a comprehensive analysis of token distributions across six diverse commonsense reasoning datasets: ARC-Challenge, ARC-Easy, BoolQ, HellaSwag, PIQA, and SIQA. These datasets span a wide range of commonsense reasoning tasks, from factual knowledge retrieval (ARC datasets) and yes/no question answering (BoolQ) to physical commonsense (PIQA) and social understanding (SIQA).

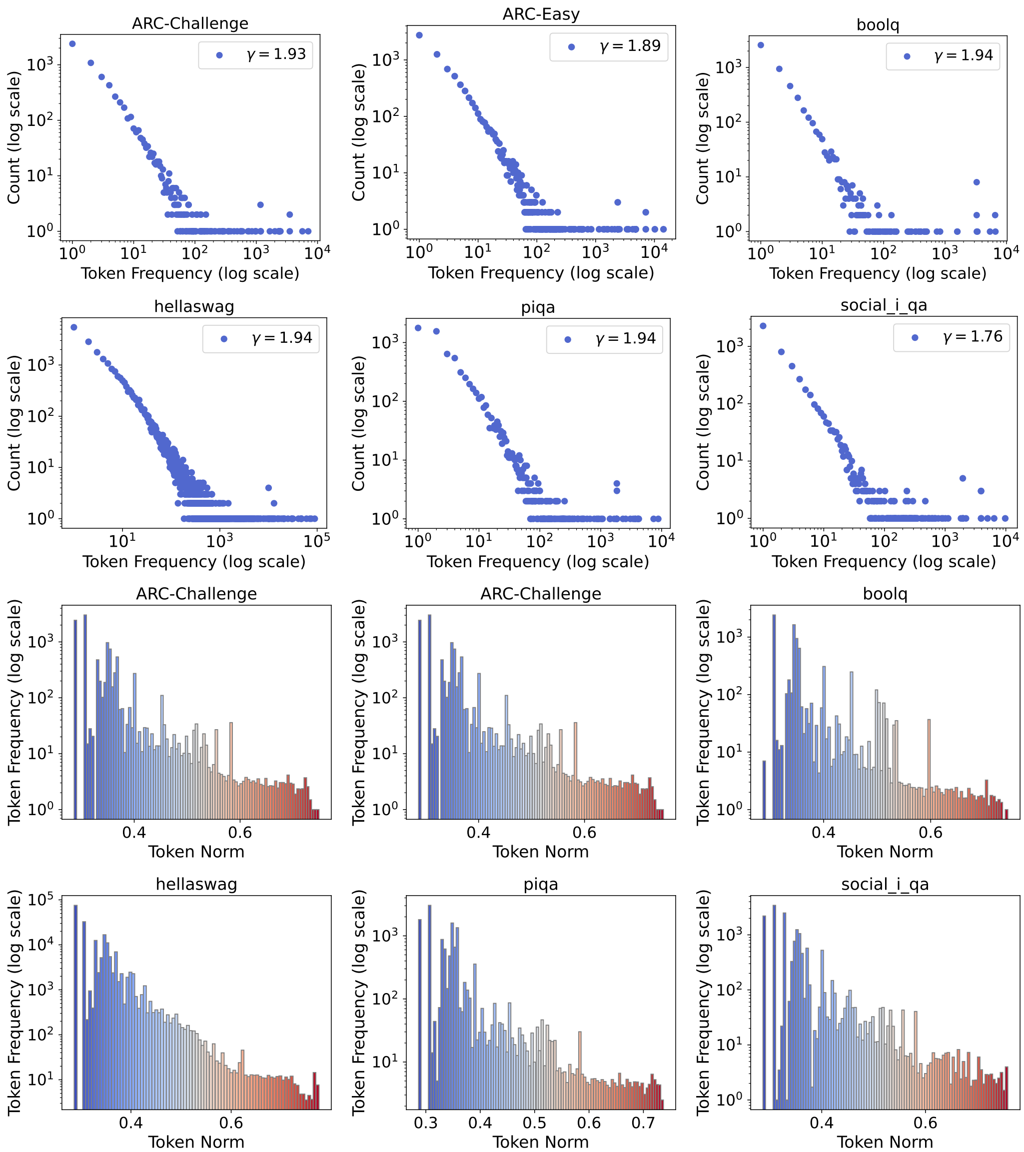


Figure 4: Token frequency distribution (top two rows) and token frequency vs. norm (bottom two rows) across different commonsense reasoning datasets in LLaMA3. The top two rows show the power-law distribution of token frequencies with the decay rate (γ) annotated for each dataset. The bottom two rows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

Table 6: Relative δ-hyperbolicity (mean $\pm$ std.) of the final hidden layer in Gemma-7B across math (AQuA, GSM8K) and commonsense (ARC-Challenge, WinoGrande, OpenBookQA) datasets comparing the frozen base model, LoRA, DoRA, and HypLoRA.

Dataset	**Base Model**	**LoRA**	**DoRA**	**HypLoRA**
AQuA	0.31 ± 0.04	0.24 ± 0.05	0.23 ± 0.05	0.22 ± 0.03
GSM8K	0.28 ± 0.04	0.21 ± 0.05	0.21 ± 0.05	0.20 ± 0.03
ARC-Challenge	0.30 ± 0.03	0.35 ± 0.03	0.36 ± 0.02	0.25 ± 0.02
Winogrande	0.22 ± 0.04	0.32 ± 0.02	0.27 ± 0.02	0.27 ± 0.02
OpenbookQA	0.30 ± 0.03	0.35 ± 0.03	0.38 ± 0.02	0.25 ± 0.02

Table 7: Relative δ-hyperbolicity (mean $\pm$ std.) of the final hidden layer in Gemma3-4B for the same five datasets, contrasting the base model with LoRA, DoRA, and HypLoRA.

Dataset	**Base Model**	**LoRA**	**DoRA**	**HypLoRA**
AQuA	0.17 ± 0.03	0.17 ± 0.03	0.19 ± 0.02	0.11 ± 0.01
GSM8K	0.16 ± 0.03	0.20 ± 0.03	0.19 ± 0.03	0.11 ± 0.02
ARC-Challenge	0.17 ± 0.02	0.21 ± 0.01	0.17 ± 0.02	0.20 ± 0.02
Winogrande	0.16 ± 0.02	0.16 ± 0.02	0.21 ± 0.01	0.12 ± 0.01
OpenbookQA	0.17 ± 0.03	0.16 ± 0.02	0.17 ± 0.03	0.11 ± 0.01

This diverse collection allows us to investigate whether the hyperbolic characteristics observed in mathematical reasoning extend to broader domains of human knowledge and reasoning. The inclusion of both challenging (ARC-Challenge, HellaSwag) and more accessible (ARC-Easy, BoolQ) datasets enables us to examine how task difficulty influences the underlying geometric structure of token embeddings.

The results presented in Figure 4 demonstrate that the power-law distribution of token frequencies and the inverse relationship between frequency and embedding norms persist across all commonsense reasoning datasets, with power-law exponents ranging from $\gamma = 1.76$ to $\gamma = 1.94$. Notably, the Social IQA dataset exhibits a slightly lower exponent ($\gamma = 1.76$), suggesting that social reasoning tasks may have a somewhat different hierarchical structure, possibly due to the more nuanced and context-dependent nature of social interactions compared to factual or physical reasoning. Despite this variation, the overall pattern remains consistent: abstract concepts and function words maintain smaller norms and higher frequencies, while specific entities, proper nouns, and domain-specific terminology are positioned at greater distances from the origin.

A.3 Hyperbolicity in the Final Hidden Layer of LLMs

In this part, we further present the analysis of the hyperbolicity of the hidden states in Tables 6 and Table 7. Considering five distinct reasoning datasets, including two mathematical reasoning datasets (AQuA and GSM8K) as well as three commonsense reasoning datasets (ARC-Challenge, Winogrande, and OpenbookQA), we observe that the base models consistently exhibit less hyperbolic structure (i.e., higher δ values) in their final hidden layer representations compared to their initial token embeddings.

LoRA and DoRA generally reduce the δ values, while the proposed HypLoRA method mostly achieves even lower values, indicating a higher degree of hyperbolicity in the learned representations. This effect is observed across most datasets in both model families. These empirical findings complement our analysis of the initial token embeddings: while the pretrained models begin with a latent hierarchical structure, as evidenced by hyperbolicity in the input layer, fine-tuning methods can either preserve or distort this property. The consistently lower δ values of HypLoRA provide strong empirical evidence that our method actively preserves and enhances the hierarchical structure of the representations throughout the model, aligning the final contextualized embeddings with the geometric biases that are beneficial for reasoning.

B Hyperbolicity on Different Metric Spaces

Table 2 presents the hyperbolicity values in both continuous (i.e., sphere space) and discrete metric spaces (i.e., tree, scale-free, and random graphs). We employ a consistent processing method similar to that used in Section 4 for embedding spaces. Specifically, we sample 1,000 four-tuples, compute the δ value for each, and then take the maximum value.

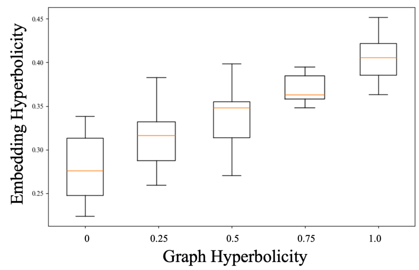


Figure 5: Empirical correlation between the ground-truth δ-hyperbolicity of several reference graphs (tree, scale-free, PubMed, dense, sphere) and the δ measured after embedding them with a two-layer GCN into Euclidean space; each point averages 1,000 sampled quadruples.

For the sphere space, we use a two-dimensional model and calculate hyperbolicity based on geodesic distances. The PubMed graph is sourced from Sen et al. [99]. The tree and dense graphs are generated using NetworkX [100]. For these graphs, we remove isolated nodes before performing our calculations to ensure consistency. We use the shortest-path distance on each graph as the distance measure, analogous to the concept of geodesics in continuous spaces.

In this study, we utilize the Euclidean distance to compute the hyperbolicity of token embeddings, following the approach proposed by [16]. To further assess the validity of this method, we embed graphs with varying degrees of hyperbolicity into Euclidean space using a two-layer GCN and compute hyperbolicity based on the distances between embeddings. The results, presented in Figure 5, indicate a positive correlation between the hyperbolicity of the original graphs and that of the embeddings, although the values do not exactly coincide. Building on this observed relationship, we calculate the hyperbolicity of token embeddings as a proxy for estimating their underlying geometric structure. In this context, lower hyperbolicity values suggest a more tree-like geometric configuration.

C Exponential and Logarithmic Maps

The exponential and logarithmic maps are fundamental tools for navigating between the tangent space and the hyperbolic manifold. These maps enable us to perform computations in the familiar Euclidean tangent space while preserving the geometric properties of hyperbolic space.

C.1 Exponential Map

The exponential map $\exp_{\mathbf{x}}^K : \mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n \to \mathcal{L}_K^n$ projects a tangent vector $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n$ at point $\mathbf{x}$ onto the hyperboloid $\mathcal{L}_K^n$. Geometrically, it maps $\mathbf{v}$ to the point $\exp_{\mathbf{x}}^K(\mathbf{v}) := \gamma(1)$, where γ is the unique geodesic satisfying $\gamma(0) = \mathbf{x}$ and $\dot{\gamma}(0) = \mathbf{v}$.

The exponential map is given by:

$$\exp_{\mathbf{x}}^K(\mathbf{v}) = \cosh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right)\mathbf{x} + \sqrt{K}\sinh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right)\frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}}, \tag{11}$$

where $\|\mathbf{v}\|_{\mathcal{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle_{\mathcal{L}}}$ is the norm of the tangent vector under the Lorentzian inner product.

At the origin $\mathbf{o} = (\sqrt{K}, 0, \ldots, 0)$, for a tangent vector $\mathbf{v} = (0, \mathbf{u})$ where $\mathbf{u} \in \mathbb{R}^n$, the exponential map simplifies to:

$$\exp_{\mathbf{o}}^K(\mathbf{v}) = \left(\sqrt{K}\cosh\left(\frac{\|\mathbf{u}\|}{\sqrt{K}}\right),\ \sqrt{K}\sinh\left(\frac{\|\mathbf{u}\|}{\sqrt{K}}\right)\frac{\mathbf{u}}{\|\mathbf{u}\|}\right). \tag{12}$$

C.2 Logarithmic Map

The logarithmic map $\log_{\mathbf{x}}^K : \mathcal{L}_K^n \to \mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n$ is the inverse of the exponential map. It projects a point $\mathbf{y} \in \mathcal{L}_K^n$ back to the tangent space at $\mathbf{x}$:

$$\log_{\mathbf{x}}^K(\mathbf{y}) = \frac{\cosh^{-1}\left(-\frac{\langle \mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}}{K}\right)}{\sqrt{\left(\frac{\langle \mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}}{K}\right)^2 - 1}} \left(\mathbf{y} + \frac{\langle \mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}}{K}\mathbf{x}\right). \tag{13}$$

These maps satisfy the inverse relationships: $\log_{\mathbf{x}}^K(\exp_{\mathbf{x}}^K(\mathbf{v})) = \mathbf{v}$ and $\exp_{\mathbf{x}}^K(\log_{\mathbf{x}}^K(\mathbf{y})) = \mathbf{y}$.

C.3 Notation in the Main Text

In the main text, we use the shorthand notation $\Pi_{\exp}^K$ and $\Pi_{\log}^K$ to denote general projection operators between Euclidean space $\mathbb{R}^n$ and hyperbolic space $\mathcal{L}_K^n$. The exponential and logarithmic maps described above represent one valid instantiation of these operators:

$$\Pi_{\exp}^K(\mathbf{x}) := \exp_{\mathbf{o}}^K((0,\mathbf{x})), \tag{14}$$

$$\Pi_{\log}^K(\mathbf{y}) := \log_{\mathbf{o}}^K(\mathbf{y})_{[1:]}, \tag{15}$$

where $(0,\mathbf{x}) \in \mathbb{R}^{1+d}$ denotes the vector obtained by prepending a zero to $\mathbf{x} \in \mathbb{R}^d$, and $(\cdot)_{[1:]}$ denotes the restriction to the last d coordinates (i.e., removal of the first coordinate). However, other diffeomorphisms [101] between Euclidean and hyperbolic spaces. The choice of projection method can be adapted based on computational efficiency and numerical stability requirements, while the core principle of our approach, performing the low-rank transformation directly on the hyperbolic manifold, remains unchanged.

Important Observation. Regardless of the specific projection method used, when these maps are applied consecutively at the same base point without intermediate operations on the manifold, they effectively cancel each other out. For example, $\log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\mathbf{v})) = \mathbf{v}$. This is why the conventional tangent-space approach for hyperbolic neural networks [33, 32] does not directly benefit LLM adaptation, where the hyperbolic geometry is effectively bypassed. Our Direct Lorentz Low-Rank Transformation (LLR) addresses this limitation by operating directly on the hyperbolic manifold between the projection steps, ensuring that the geometric properties of hyperbolic space are preserved and utilized.

D Lorentz Transformation

In the context of special relativity, Lorentz transformations are linear mappings that preserve the spacetime interval between events, ensuring the constancy of the speed of light across all inertial frames. These transformations can be categorized into two primary types: Lorentz boosts and Lorentz rotations [102, 103].

D.1 Lorentz Boost

A Lorentz boost corresponds to a transformation between two inertial reference frames moving at a constant relative velocity. Given a velocity vector $\mathbf{v} \in \mathbb{R}^n$ with magnitude $\|\mathbf{v}\| < 1$, the Lorentz boost matrix $\mathbf{B}$ mixes time and space coordinates:

$$\mathbf{B} = \begin{bmatrix} \gamma & -\gamma\mathbf{v}^\top \\ -\gamma\mathbf{v} & \mathbf{I} + \frac{\gamma^2}{1+\gamma}\mathbf{v}\mathbf{v}^\top \end{bmatrix}, \tag{16}$$

where $\gamma = \frac{1}{\sqrt{1-\|\mathbf{v}\|^2}}$ is the Lorentz factor.

D.2 Lorentz Rotation

A Lorentz rotation involves only the rotation of spatial coordinates while preserving the time coordinate:

$$\mathbf{R} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \tilde{\mathbf{R}} \end{bmatrix}, \tag{17}$$

where $\tilde{\mathbf{R}} \in SO(n)$ is a spatial rotation matrix.

Our Spatial-like Transformation. In our Direct Lorentz Low-Rank Transformation (LLR), we apply transformations exclusively to the spatial components while maintaining the constraint of the Lorentz manifold. Given a point $\mathbf{x}^H = (x_0^H, \mathbf{x}_s^H) \in \mathcal{L}_K^n$, our transformation is:

$$\mathbf{LLR}(BA, \mathbf{x}^H) = (\sqrt{\|BA\mathbf{x}_s^H\|^2 + K}, BA\mathbf{x}_s^H), \tag{18}$$

where we transform the spatial component $\mathbf{x}_s^H$ and recompute the time component to maintain the Lorentz constraint $x_0^2 - \|\mathbf{x}_s\|^2 = K$.

This can be decomposed into two sequential transformations:

$$\mathbf{y}^H = (y_0^H, \mathbf{y}_s^H) = (\sqrt{\|A\mathbf{x}_s^H\|^2 + K}, A\mathbf{x}_s^H), \tag{19}$$

$$\mathbf{z}^H = (z_0^H, \mathbf{z}_s^H) = (\sqrt{\|B\mathbf{y}_s^H\|^2 + K}, B\mathbf{y}_s^H). \tag{20}$$

Interpretation as a Constrained Lorentz Rotation. Our transformation can be viewed as a special case of Lorentz rotation where: (1) We apply a linear transformation to the spatial coordinates: $\mathbf{x}_s^H \mapsto BA\mathbf{x}_s^H$; (2) We recompute the time component to preserve the manifold constraint: $x_0^H \mapsto \sqrt{\|BA\mathbf{x}_s^H\|^2 + K}$. This approach differs from a standard Lorentz rotation in two ways (see also [37]): (1) the spatial transformation BA is not necessarily orthogonal (i.e., $BA \notin SO(n)$); (2) the time component is not preserved but rather recomputed to maintain the manifold constraint.

In matrix form, our transformation can be expressed as:

$$\begin{bmatrix} z_0^H \\ \mathbf{z}_s^H \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{\|BA\mathbf{x}_s^H\|^2+K}}{\sqrt{\|\mathbf{x}_s^H\|^2+K}} & \mathbf{0}^\top \\ \mathbf{0} & BA \end{bmatrix} \begin{bmatrix} x_0^H \\ \mathbf{x}_s^H \end{bmatrix} \tag{21}$$

The key property is that this transformation preserves the Lorentz manifold structure: if $\mathbf{x}^H \in \mathcal{L}_K^n$, then $\mathbf{LLR}(BA, \mathbf{x}^H) \in \mathcal{L}_K^n$, as verified by:

$$(z_0^H)^2 - \|\mathbf{z}_s^H\|^2 = \|BA\mathbf{x}_s^H\|^2 + K - \|BA\mathbf{x}_s^H\|^2 = K. \tag{22}$$

This spatial-like transformation approach allows us to leverage the low-rank structure of BA while maintaining the geometric properties of the hyperbolic space, providing a computationally efficient method for hyperbolic low-rank adaptation.

E Transformation Analysis

This section provides a detailed analysis of how HypLoRA differs from standard LoRA by examining the higher-order terms introduced through hyperbolic geometry.

Proof. Let $\mathbf{x} \in \mathbb{R}^d$ be an input token embedding. Let $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{k \times r}$ be low-rank matrices with rank $r \ll \min\{d, k\}$. Consider the d-dimensional hyperbolic space $\mathcal{L}_K^d$ (Lorentz model) with curvature $C = -1/K$, where $K > 0$.

Our goal is to analyze how the HypLoRA update differs from the LoRA update and to understand the impact of token norms $\|\mathbf{x}\|$ on the higher-order terms introduced by HypLoRA.

Mapping the Input Embedding to Hyperbolic Space. Following previous work [32], we interpret the Euclidean token embedding $\mathbf{x}$ as an element in the tangent space at the origin $\mathbf{o}$ of the hyperbolic space $\mathcal{L}_K^d$. The tangent vector is given by $\mathbf{v} = (0, \mathbf{x}) \in T_{\mathbf{o}}\mathcal{L}_K^d$. The exponential map $\exp_{\mathbf{o}}^K$ projects $\mathbf{v}$ onto the hyperbolic space:

$$\exp_{\mathbf{o}}^K(\mathbf{v}) = \left(\sqrt{K}\cosh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right),\ \sqrt{K}\sinh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right)\frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}}\right), \tag{23}$$

where $\|\mathbf{v}\|_{\mathcal{L}}$ denotes the Minkowski norm. Since $\mathbf{v} = (0, \mathbf{x})$ and $\|\mathbf{v}\|_{\mathcal{L}} = \|\mathbf{x}\|$, the exponential map simplifies to:

$$\exp_{\mathbf{o}}^K(\mathbf{v}) = \left(\sqrt{K}\cosh\left(\frac{\|\mathbf{x}\|}{\sqrt{K}}\right),\ \sqrt{K}\sinh\left(\frac{\|\mathbf{x}\|}{\sqrt{K}}\right)\frac{\mathbf{x}}{\|\mathbf{x}\|}\right). \tag{24}$$

Approximations. For small $\frac{\|\mathbf{x}\|}{\sqrt{K}}$, let $z = \|\mathbf{x}\|$ we can use the Taylor series expansions:

$$\cosh\left(\frac{z}{\sqrt{K}}\right) \approx 1 + \frac{z^2}{2K}, \quad \sinh\left(\frac{z}{\sqrt{K}}\right) \approx \frac{z}{\sqrt{K}} + \frac{z^3}{6K^{3/2}}. \tag{25}$$

Applying these to the exponential map of $\mathbf{x}$:

$$u_0^H \approx \sqrt{K} + \frac{\|\mathbf{x}\|^2}{2\sqrt{K}}, \tag{26}$$

$$\mathbf{u}_{\text{space}}^H \approx \mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}\mathbf{x}. \tag{27}$$

Applying Low-Rank Transformations to the Approximated Embedding. Using the approximated $\mathbf{u}_{\text{space}}^H$, we apply the transformations.

First transformation:

$$\mathbf{y}_{\text{space}}^H = A\mathbf{u}_{\text{space}}^H \approx A\left(\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}\mathbf{x}\right) = A\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}A\mathbf{x}. \tag{28}$$

Second transformation:

$$\mathbf{z}_{\text{space}}^H = B\mathbf{y}_{\text{space}}^H \approx BA\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \tag{29}$$

Compute the time component after the transformations:

$$z_0^H = \sqrt{K + \left\|\mathbf{z}_{\text{space}}^H\right\|^2}. \tag{30}$$

Approximating the Logarithmic Map. We map the transformed hyperbolic point $\mathbf{z}^H = \left(z_0^H, \mathbf{z}_{\text{space}}^H\right)$ back to the tangent space at the origin using the logarithmic map $\log_{\mathbf{o}}^K$:

$$\Delta Q^{\text{Hyp}} = \log_{\mathbf{o}}^K(\mathbf{z}^H) = \sqrt{K} \cdot \operatorname{arcosh}\left(\frac{z_0^H}{\sqrt{K}}\right)\frac{\mathbf{z}_{\text{space}}^H}{\sqrt{(z_0^H)^2 - K}}. \tag{31}$$

Using the approximation $z_0^H \approx \sqrt{K} + \frac{\left\|\mathbf{z}_{\text{space}}^H\right\|^2}{2\sqrt{K}}$ and for small $\delta = \frac{\left\|\mathbf{z}_{\text{space}}^H\right\|^2}{2K}$, we have:

$$\operatorname{arcosh}\left(\frac{z_0^H}{\sqrt{K}}\right) \approx \operatorname{arcosh}(1+\delta) \approx \sqrt{2\delta} = \frac{\left\|\mathbf{z}_{\text{space}}^H\right\|}{\sqrt{K}}, \tag{32}$$

$$\sqrt{(z_0^H)^2 - K} \approx \left\|\mathbf{z}_{\text{space}}^H\right\|. \tag{33}$$

Therefore, the logarithmic map simplifies to:

$$\Delta Q^{\text{Hyp}} \approx \mathbf{z}_{\text{space}}^H. \tag{34}$$

Comparing HypLoRA and LoRA Updates. The HypLoRA update is:

$$\Delta Q^{\text{Hyp}} \approx BA\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \tag{35}$$

The LoRA update is:

$$\Delta Q^{\text{LoRA}} = BA\mathbf{x}. \tag{36}$$

The difference between the updates is:

$$\Delta Q^{\text{Hyp}} - \Delta Q^{\text{LoRA}} = \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \tag{37}$$

Impact of Token Norms on Higher-Order Terms. The higher-order term $\frac{\|\mathbf{x}\|^2}{6K} BA\mathbf{x}$ is proportional to $\|\mathbf{x}\|^2$. Since $\|\mathbf{x}\|$ reflects the specificity of the token in the hierarchical structure (larger norms correspond to more specific tokens), this term becomes significant for tokens representing specific concepts.

Impact on Attention Scores. The HypLoRA attention scores are computed as:

$$\text{Scores}_{\text{HypLoRA}} = \frac{\left(Q^{\text{orig}} + \Delta Q^{\text{Hyp}}\right)\left(K^{\text{orig}} + \Delta K^{\text{Hyp}}\right)^{\top}}{\sqrt{d_k}}, \tag{38}$$

where ΔK^{Hyp} is derived similarly.

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

$$\Delta\,\text{Scores} = \text{Scores}_{\text{HypLoRA}} - \text{Scores}_{\text{LoRA}}\,. \tag{39}$$

These higher-order terms enable HypLoRA to capture more complex hierarchical relationships, particularly for tokens with larger norms.

□

Remark 1. ***Alignment with Token Hierarchy****: The higher-order terms in HypLoRA's updates are proportional to $\|\mathbf{x}\|^2$, correlating with the specificity of tokens in the hierarchical structure. As a result, HypLoRA places greater emphasis on more specific tokens, enhancing its ability to model detailed relationships.*

***Role of Curvature** C: The curvature $C = -1/K$ scales the higher-order corrections. Smaller K (larger negative curvature) amplifies these terms, aligning with the hyperbolic nature of token embeddings. In practice, the curvature parameter K can be tuned to ensure this condition is satisfied for typical token embedding norms.*

***Effectiveness of HypLoRA**: By incorporating these higher-order terms, HypLoRA leverages the inherent hierarchical and hyperbolic structure of token embeddings. This leads to improved performance, especially on problems requiring complex reasoning, explaining why the proposed method performs better on more challenging datasets.*

F Full Comparison

While the main body of our paper focuses on comparing HypLoRA against the standard LoRA baseline to demonstrate the core effectiveness of our hyperbolic fine-tuning approach, this section provides a comprehensive evaluation against a broader range of parameter-efficient fine-tuning methods, such as Prefix tuning [68], Series and Parallel adapters [70], and DoRA [75], providing a more complete picture of HypLoRA's performance relative to the current landscape of efficient fine-tuning techniques. This extended comparison validates that our improvements are not merely due to increased model capacity or specific architectural choices, but rather stem from the fundamental advantages of incorporating hyperbolic geometry into the adaptation process.

F.1 Implementation Details

To ensure consistency and comparability, our experimental setup closely followed the training configurations outlined in Hu et al. [98]. Across all fine-tuning tasks, we employed the AdamW optimizer with a learning rate of 3×10^{-4} and trained for a total of three epochs. LoRA modules (and consequently, HypLoRA adapters) were integrated into both the Multi-Head Attention (MHA) and MLP layers of the foundation models. A key hyperparameter for HypLoRA is the curvature K (defining the hyperbolic curvature as $-1/K$), which was initialized by searching the set $\{0.5, 1.0\}$. For evaluation, final scores were micro-averaged for arithmetic reasoning and averaged for commonsense reasoning across the datasets, thereby giving equal weight to each individual prompt, regardless of the varying number of questions per dataset (e.g., $1,319$ in GSM8K versus 238 in MAWPS).

For baseline methods, we adopted the following approach: results for Prefix tuning [68], Series adapters, and Parallel adapters [70] are directly cited from Hu et al. [98] to ensure fair comparison

Table 8: Comprehensive comparison of parameter-efficient fine-tuning methods on mathematical reasoning tasks. Results marked with * are from [98], while † indicates our reproduced results. The percentage following each dataset name indicates the proportion of prompts relative to the total number of inference prompts. M.AVG represents the micro-average accuracy across all datasets. Best results for each model are highlighted in bold. OOT indicates out-of-time during training.

Base Model	PEFT Method	MAWPS(8.5%)	SVAMP(35.6%)	GSM8K(46.9%)	AQuA(9.0%)	M.AVG
GPT-3.5	None	87.4	69.9	56.4	38.9	62.3
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
	HypLoRA (Ours)	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
	HypLoRA (Ours)	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	90.7	79.2	68.3	33.9	71.0
	HypLoRA (Ours)	89.5	78.7	69.5	32.7	**71.2**
LLaMA3-8B	None	79.8	50.0	54.7	21.0	52.1
	LoRA	92.7	78.9	70.8	30.4	71.9
	DoRA	90.3	79.8	73.3	21.3	72.4
	HypLoRA (Ours)	91.6	80.5	74.0	34.2	**74.2**
Gemma3-4B	LoRA	90.8	77.3	72.3	50.8	73.7
	DoRA	89.5	78.8	68.5	52.4	72.5
	HypLoRA (Ours)	88.2	83.9	76.1	53.2	**77.8**
Qwen2.5-7B	LoRA	90.8	84.4	78.6	68.1	80.8
	DoRA	92.8	87.4	80.4	64.2	82.5
	HypLoRA (Ours)	91.2	92.2	87.9	71.6	**88.3**

under identical experimental conditions. For LoRA and DoRA, we conducted independent reimplementations following their respective original papers and parameters [31, 75] to enable rigorous and controlled comparisons.

F.2 Comparison on Mathematical Reasoning

Looking at the mathematical reasoning comparison table, several key experimental findings emerge regarding HypLoRA's performance across different model architectures and datasets. The results demonstrate that HypLoRA consistently outperforms standard LoRA across multiple model families, with particularly notable improvements on more challenging datasets. For the Gemma-7B model, HypLoRA achieves a micro-averaged accuracy of 71.2%, surpassing LoRA's 68.6%. For LLaMA3-8B, HypLoRA reaches 74.2% compared to LoRA's 71.9%. The improvements are especially pronounced on the AQuA dataset, which requires complex algebraic reasoning. Specifically, HypLoRA shows gains of 3.8 percentage points over LoRA on Gemma-7B (32.7% vs 28.9%) and 3.8 points on LLaMA3-8B (34.2% vs 30.4%). This pattern suggests that HypLoRA's hyperbolic geometry is particularly effective for problems requiring multi-step reasoning and understanding of hierarchical mathematical relationships.

The consistency of improvements across different model architectures further validates the generalizability of the hyperbolic approach. While HypLoRA shows competitive performance on simpler datasets like MAWPS, the performance advantages become more significant on challenging datasets like GSM8K and AQuA, which demand sophisticated reasoning capabilities. For instance, on GSM8K, HypLoRA achieves 69.5% accuracy on Gemma-7B versus 66.3% for LoRA, and 74.0%

Table 9: Extended commonsense reasoning accuracy (%) for GPT-3.5 and for LoRA, DoRA, and HypLoRA on LLaMA3-8B, Gemma3-4B, and Qwen2.5-7B. Columns correspond to BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-e, ARC-c, and OBQA; the rightmost column reports the macro average across the eight benchmarks.

Base Model	PEFT Method	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	AVG
GPT-3.5	None	None	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA3-8B	LoRA	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA	0.71	72.1	85.5	79.6	92.8	83.3	85.2	72.1	84.0	81.8
	HypLoRA (Ours)	0.70	**74.1**	**87.6**	**80.6**	**94.5**	**84.7**	**90.4**	**81.2**	**85.2**	**84.8**
Gemma3-4B	LoRA	1.04	68.1	83.2	77.2	88.9	**80.5**	84.5	69.9	83.6	79.5
	DoRA	1.05	68.1	84.3	78.4	88.3	80.1	84.1	70.8	83.8	79.7
	HypLoRA (Ours)	1.04	**70.0**	**84.3**	**79.2**	**91.5**	80.3	**89.1**	**75.9**	**86.4**	**82.5**
Qwen2.5-7B	LoRA	0.71	**73.4**	**89.5**	79.5	93.6	84.1	92.8	82.0	87.0	85.2
	DoRA	0.72	71.7	88.7	79.0	93.7	84.1	92.4	82.8	88.4	85.1
	HypLoRA (Ours)	0.71	72.8	89.3	**79.8**	**94.8**	**84.4**	**95.5**	**87.5**	**90.8**	**87.0**

on LLaMA3-8B versus 70.8% for LoRA. These correspond to gains of 3.2 points over LoRA on both Gemma-7B and LLaMA3-8B. Notably, on the most recent models, HypLoRA demonstrates substantial improvements: on Gemma3-4B, HypLoRA achieves 77.8% M.AVG compared to LoRA's 73.7% (+4.1 points), and on Qwen2.5-7B, HypLoRA reaches 88.3% versus LoRA's 80.8% (+7.5 points). The fact that HypLoRA maintains superior performance across both older (LLaMA-7B, LLaMA-13B) and newer (LLaMA3-8B, Gemma3-4B, Qwen2.5-7B) model architectures demonstrates the robustness of incorporating hyperbolic inductive biases into parameter-efficient fine-tuning, regardless of the underlying model's specific architectural details or training paradigms.

F.3 Comparison on Commonsense Reasoning

HypLoRA demonstrates substantial improvements over standard LoRA across diverse commonsense reasoning benchmarks, as shown in Table 9. The commonsense reasoning tasks evaluated include BoolQ (yes/no question answering), PIQA (physical commonsense inference), SIQA (social interaction reasoning), HellaSwag (commonsense natural language inference), WinoGrande (pronoun disambiguation), ARC-e and ARC-c (science question answering with easy and challenging difficulty), and OBQA (open book question answering). These benchmarks collectively assess the model's ability to understand implicit knowledge, contextual nuances, and real-world reasoning patterns. The consistent performance gains across all three model architectures and eight diverse benchmarks indicate that the hierarchical inductive bias introduced by hyperbolic geometry effectively captures the implicit relational structures underlying commonsense reasoning

F.4 GPU Usage

Table 10 presents a comprehensive comparison of memory usage across different fine-tuning methods for both LLaMA3-8B and Gemma3-4B models. The results demonstrate that HypLoRA maintains comparable memory efficiency to the baseline LoRA method. Specifically, HypLoRA with stereographic projection achieves identical memory consumption to LoRA (30.12 GB for LLaMA3-8B and 14.61 GB for Gemma3-4B), while HypLoRA with exponential/logarithmic maps introduces only a minimal overhead of 0.02 GB for LLaMA3-8B and 0.01 GB for Gemma3-4B. Notably, both HypLoRA variants significantly outperform DoRA in terms of memory efficiency, with DoRA requiring 30.23 GB and 14.62 GB, respectively. These results confirm that our hyperbolic adaptation approach does not compromise memory efficiency while delivering superior performance improvements, making HypLoRA a practical choice for resource-constrained environments where both performance gains and memory conservation are critical considerations.

Table 10: Allocated Memory Usage Comparison for Fine-tuning Methods

	LLaMA3-8B		Gemma3-4B	
Method	**Param (%)**	**Memory (GB)**	**Param (%)**	**Memory (GB)**
LoRA	0.7002	30.12	1.0371	14.61
DoRA	0.7098	30.23	1.0516	14.62
HypLoRA	0.7099	30.14	1.0518	14.62

Table 11: Example problem with solutions from LoRA and HypLoRA (Case 1)

Question	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? **Answer:** 80.0
LoRA	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. **Answer: 76**.
HypLoRA	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. **Answer: 80**.

G Case Study

In this section, we present three examples to demonstrate the advantages of incorporating HypLoRA over standard LoRA fine-tuning.

Table 12: Example problem with solutions from LoRA and HypLoRA (Case 2)

Question	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? **Answer:** 20.0 minutes
LoRA	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. **Answer: 60**.
HypLoRA	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. **Answer: 20**.

Table 13: Example problem with solutions from LoRA and HypLoRA (Case 3)

Question	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? **Answer:** 2,800 pounds
LoRA	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. **Answer: 1,800**.
HypLoRA	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. **Answer: 2,800**.

These examples demonstrate how HypLoRA consistently provides more accurate reasoning compared to LoRA across different types of mathematical problems. In Case 1, LoRA drops 20 points when summing the five scores (reporting 380 instead of 400) and therefore produces the wrong average. This seemingly small arithmetic lapse aligns with the observation that LLMs often rely on high-level pattern similarity rather than exact computation [104]. By preserving greater separation among numerically close but semantically distinct tokens (e.g., 380 vs. 400), the hyperbolic representation in HypLoRA keeps the sequence of operations faithful and recovers the correct average.

In Case 2, LoRA immediately divides 240 minutes by the four weekly walking days, yielding 60 minutes per day and ignoring that the 240-minute budget must be spread over three weeks. HypLoRA correctly reasons in stages: divide 240 by 3 weeks, then by 4 days per week, recovering the required 20 minutes per day and showing stronger temporal reasoning.

In Case 3, LoRA actually sets up the correct balance equation $4{,}000+4{,}800+x = 11{,}600$ but subtracts 9,800 from 11,600 rather than 8,800, reporting $x = 1{,}800$. HypLoRA carries the subtraction through correctly and outputs the true 2,800 pounds. Together, these examples illustrate how the hyperbolic geometry employed by HypLoRA enables better handling of multi-step reasoning, maintaining both semantic context and numerical consistency in mathematical problem-solving scenarios.

Overall, these cases highlight a consistent trend: LoRA frequently derails on either a single arithmetic step (Cases 1 and 3) or a latent multi-hop dependency (Case 2), whereas HypLoRA preserves each intermediate calculation, keeps quantities well separated in representation space, and consequently delivers the correct final answers. These qualitative observations complement the quantitative gains reported in the main paper.