

## Appendix Contents

<b>A</b>	<b>More Investigation Results</b>	<b>25</b>
A.1	Token Frequency and Norm Distribution on Mathematical Reasoning . . . . .	25
A.2	Token Frequency and Norm Distribution on Commonsense Reasoning . . . . .	25
A.3	Hyperbolicity in the Final Hidden Layer of LLMs . . . . .	27
<b>B</b>	<b>Hyperbolicity on Different Metric Spaces</b>	<b>27</b>
<b>C</b>	<b>Lorentz Transformation</b>	<b>28</b>
<b>D</b>	<b>Projections Between Euclidean and Hyperbolic Space</b>	<b>29</b>
D.1	Exponential and Logarithmic Map . . . . .	29
D.2	Stereographic Projection . . . . .	30
D.3	Comparisons . . . . .	31
<b>E</b>	<b>Transformation Analysis</b>	<b>31</b>
E.1	Transformation by Exponential and Logarithmic Mappings . . . . .	31
E.2	Transformation by Stereographic Projection . . . . .	33
E.3	Difference with Hyperbolic Dimension Reduction Methods . . . . .	34
<b>F</b>	<b>Full Comparison</b>	<b>35</b>
F.1	Comparison on Mathematical Reasoning . . . . .	35
F.2	Comparison on Commonsense Reasoning . . . . .	36
F.3	GPU Usage . . . . .	36
<b>G</b>	<b>Case Study</b>	<b>36</b>

## 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.

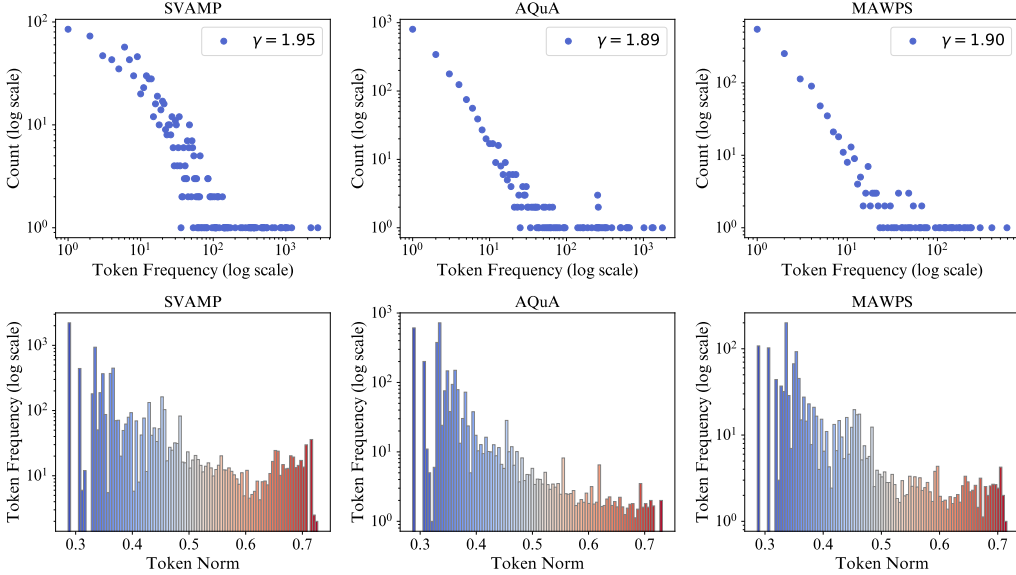


Figure 3: Token frequency distribution (top row) and token frequency vs. norm (bottom row) across different mathematical reasoning datasets in LLaMA3. The top row shows the power-law distribution of token frequencies with the decay rate ( $\gamma$ ) annotated for each dataset. The bottom row illustrates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

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 seven diverse commonsense reasoning datasets: ARC-Challenge, ARC-Easy, BoolQ, HellaSwag, OBQA, 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

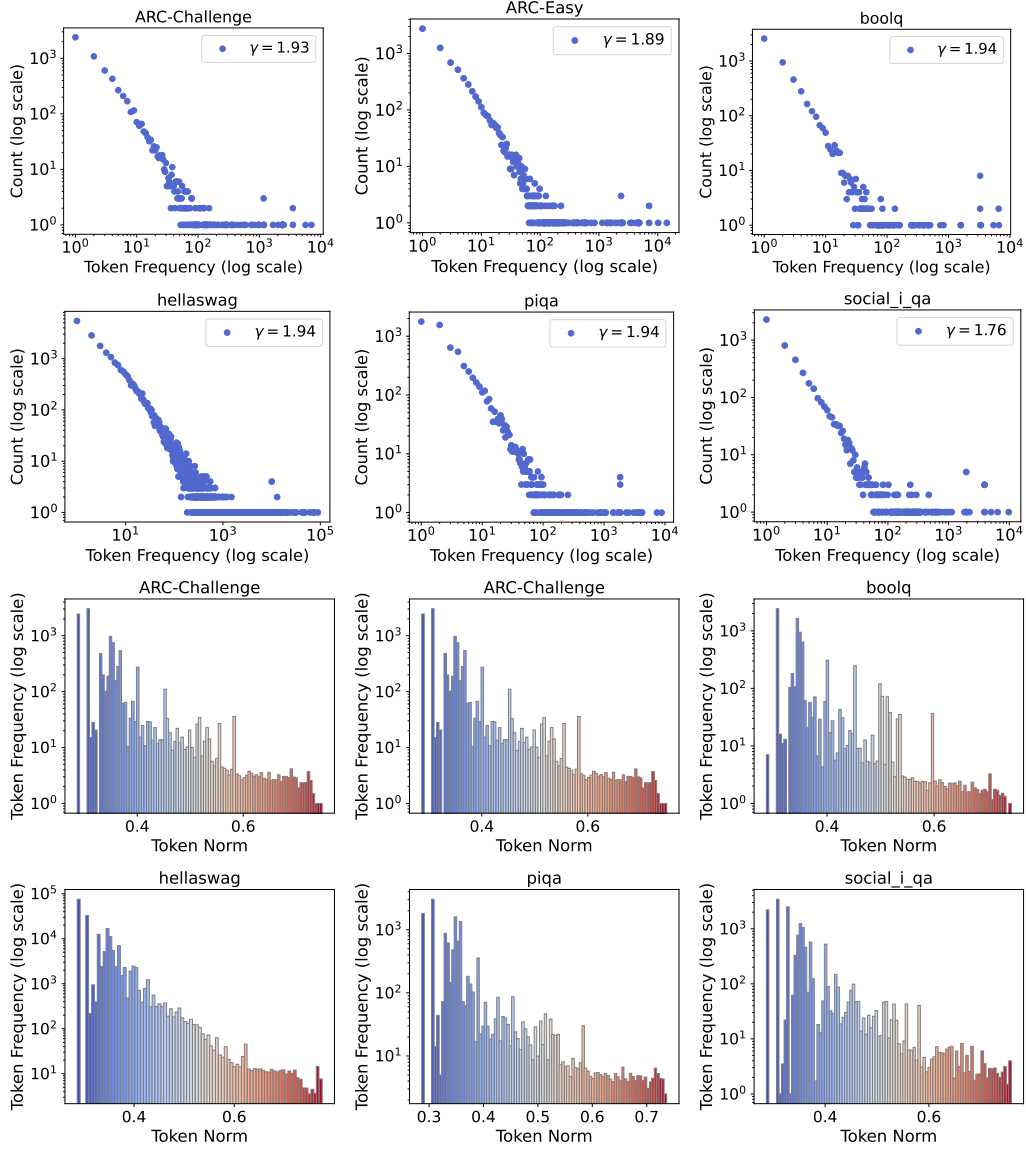


Figure 4: Token frequency distribution (top two rows) and token frequency vs. norm (bottom two rows) across different commonsense reasoning datasets in LLaMA3. The top two rows show the power-law distribution of token frequencies with the decay rate ( $\gamma$ ) annotated for each dataset. The bottom 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: Hyperbolicity of Final Hidden Layer Embeddings in Gemma 7B

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

Table 7: Hyperbolicity of Final Hidden Layer Embeddings in Gemma3-4B

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

(SIQA). 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 a lower degree of hyperbolicity (i.e., higher hyperbolicity values) in their final hidden layer representations compared to their initial token embeddings.

LoRA and DoRA generally reduce the hyperbolicity 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  $\delta$  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  $\delta$  value for each, and then take the maximum value.

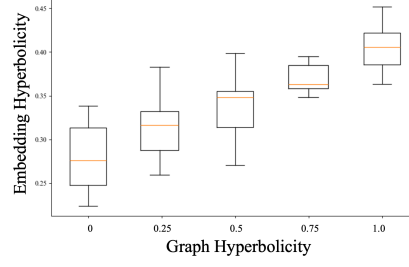


Figure 5: Correlation between graph and embedding hyperbolicity.

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. [94]. The tree and dense graphs are generated using NetworkX [95]. 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 [14]. 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 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 [96, 97].

**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:

$$\mathbf{B} = \begin{bmatrix} \gamma & -\gamma \mathbf{v}^\top \\ -\gamma \mathbf{v} & \mathbf{I} + \frac{\gamma^2}{1+\gamma} \mathbf{v} \mathbf{v}^\top \end{bmatrix}, \quad (10)$$

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

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

$$\mathbf{R} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \tilde{\mathbf{R}} \end{bmatrix}, \quad (11)$$

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:

$$\text{LLR}(\mathbf{B}\mathbf{A}, \mathbf{x}^H) = (\sqrt{\|\mathbf{B}\mathbf{A}\mathbf{x}_s^H\|^2 + K}, \mathbf{B}\mathbf{A}\mathbf{x}_s^H), \quad (12)$$

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:

$$\mathbf{y}^H = (y_0^H, \mathbf{y}_s^H) = (\sqrt{\|\mathbf{A}\mathbf{x}_s^H\|^2 + K}, \mathbf{A}\mathbf{x}_s^H), \quad (13)$$

$$\mathbf{z}^H = (z_0^H, \mathbf{z}_s^H) = (\sqrt{\|\mathbf{B}\mathbf{y}_s^H\|^2 + K}, \mathbf{B}\mathbf{y}_s^H). \quad (14)$$

**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:  $\mathbf{x}_s^H \mapsto \mathbf{B}\mathbf{A}\mathbf{x}_s^H$ ; (2) We recompute the time component to preserve the manifold constraint:  $x_0^H \mapsto \sqrt{\|\mathbf{B}\mathbf{A}\mathbf{x}_s^H\|^2 + K}$ . This approach differs from a standard Lorentz rotation in two ways (see also [35]): (1) the spatial transformation  $\mathbf{B}\mathbf{A}$  is not necessarily orthogonal (i.e.,  $\mathbf{B}\mathbf{A} \notin \text{SO}(n)$ ); (2) the time component is not preserved but rather recomputed to maintain the manifold constraint.

In matrix form, our transformation can be expressed as:

$$\begin{bmatrix} z_0^H \\ \mathbf{z}_s^H \end{bmatrix} = \begin{bmatrix} \sqrt{\|\mathbf{B}\mathbf{A}\mathbf{x}_s^H\|^2 + K} & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{B}\mathbf{A} \end{bmatrix} \begin{bmatrix} x_0^H \\ \mathbf{x}_s^H \end{bmatrix} \quad (15)$$

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

$$(z_0^H)^2 - \|\mathbf{z}_s^H\|^2 = \|\mathbf{B}\mathbf{A}\mathbf{x}_s^H\|^2 + K - \|\mathbf{B}\mathbf{A}\mathbf{x}_s^H\|^2 = K. \quad (16)$$

This spatial-like transformation approach allows us to leverage the low-rank structure of  $\mathbf{B}\mathbf{A}$  while maintaining the geometric properties of the hyperbolic space, providing a computationally efficient method for hyperbolic low-rank adaptation.

## D Projections Between Euclidean and Hyperbolic Space

Projecting between Euclidean and hyperbolic spaces facilitates both the transfer of standard neural network operations into hyperbolic geometry and the initialization of hyperbolic parameters from familiar Euclidean distributions.

### D.1 Exponential and Logarithmic Map

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

$$\exp_\mathbf{x}^K(\mathbf{v}) = \cosh\left(\frac{\|\mathbf{v}\|_\mathcal{L}}{\sqrt{K}}\right) \mathbf{x} + \sqrt{K} \sinh\left(\frac{\|\mathbf{v}\|_\mathcal{L}}{\sqrt{K}}\right) \frac{\mathbf{v}}{\|\mathbf{v}\|_\mathcal{L}}, \quad (17)$$

where  $\cosh$  and  $\sinh$  represent the hyperbolic cosine and sine functions, respectively, and  $\|\mathbf{v}\|_\mathcal{L}$  denotes the norm of the tangent vector  $\mathbf{v}$  in the tangent space. The logarithmic map  $\log_\mathbf{x}^K(\mathbf{y}) : \mathcal{L}_K^n \rightarrow \mathcal{T}_\mathbf{x}\mathcal{L}_K^n$  plays an inverse role. It is defined by the equation:

$$\log_\mathbf{x}^K(\mathbf{y}) = \frac{\cosh^{-1}(K\langle \mathbf{x}, \mathbf{y} \rangle_\mathcal{L})}{\sinh(\cosh^{-1}(K\langle \mathbf{x}, \mathbf{y} \rangle_\mathcal{L}))} (\mathbf{y} - K\langle \mathbf{x}, \mathbf{y} \rangle_\mathcal{L} \mathbf{x}). \quad (18)$$

The exponential and logarithmic maps establish a bijective projection between the tangent space and hyperbolic space. Notably,  $\log_\mathbf{x}^K(\exp_\mathbf{x}^K(\mathbf{v})) = \mathbf{v}$  and  $\exp_\mathbf{x}^K(\log_\mathbf{x}^K(\mathbf{y})) = \mathbf{y}$ . **Consequently, the tangent space method will cancel out the hyperbolic operations.** In addition, these operations are typically defined locally. However, in the context of hyperbolic representation and deep learning, for efficient computation, existing works usually use the origin point  $\mathbf{o} := \{\sqrt{K}, 0, \dots, 0\} \in \mathcal{L}_K^n$  as a common reference point.

## D.2 Stereographic Projection

Stereographic projection provides a smooth bijection between the Poincaré ball and Lorentz models using only elementary arithmetic. We rely on it to shuttle embeddings between Euclidean and hyperbolic spaces without the paired  $\exp^K / \log^K$  operations that cancel each other out in the tangent-space pipeline. The result is a cheaper drop-in replacement that keeps the geometry we need for HypLoRA while avoiding hyperbolic trigonometric functions that are prone to numerical overflow.

Consider the Lorentz model

$$\mathcal{L}_K^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -K, x_0 > 0\}, \quad (19)$$

embedded in Minkowski space with inner product  $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i$ . Its spatial component is denoted by  $\mathbf{x}_{1:n} = (x_1, \dots, x_n)$ . The corresponding Poincaré ball of curvature  $-1/K$  is

$$\mathbb{B}_K^n = \{\mathbf{u} \in \mathbb{R}^n : K \|\mathbf{u}\|_2^2 < 1\}, \quad (20)$$

and we use the origin  $\mathbf{o} = (\sqrt{K}, 0, \dots, 0)$  as the reference point shared across models.

**Forward map** ( $\mathcal{L}_K^n \rightarrow \mathbb{B}_K^n$ ). Dropping the point  $\mathbf{o}$  orthogonally onto the hyperplane  $x_0 = -\sqrt{K}$  yields the familiar stereographic projection

$$\Pi_{H \rightarrow B}^K(\mathbf{x}) = \frac{\mathbf{x}_{1:n}}{x_0 + \sqrt{K}}, \quad \mathbf{x} \in \mathcal{L}_K^n. \quad (21)$$

Because  $x_0 > 0$  on the upper sheet of the hyperboloid, the denominator is always positive.

**Inverse map** ( $\mathbb{B}_K^n \rightarrow \mathcal{L}_K^n$ ). Given  $\mathbf{u} \in \mathbb{B}_K^n$ , the inverse stereographic projection recovers the Lorentz coordinates as

$$\Pi_{B \rightarrow H}^K(\mathbf{u}) = \left( \sqrt{K} \frac{1 + K \|\mathbf{u}\|_2^2}{1 - K \|\mathbf{u}\|_2^2}, \frac{2\sqrt{K} \mathbf{u}}{1 - K \|\mathbf{u}\|_2^2} \right), \quad (22)$$

which satisfies  $\Pi_{H \rightarrow B}^K(\Pi_{B \rightarrow H}^K(\mathbf{u})) = \mathbf{u}$  and  $\Pi_{B \rightarrow H}^K(\Pi_{H \rightarrow B}^K(\mathbf{x})) = \mathbf{x}$ .

**Convenient Euclidean interface.** LLM hidden states live in  $\mathbb{R}^n$  and must be shrunk into  $\mathbb{B}_K^n$  before applying (22). We therefore scale each vector by a factor that is big enough to place it inside the ball to simplify the process and reduce the usage of  $\tanh$ ,  $\operatorname{arctanh}$ :

$$\gamma > \max\left(1, \sqrt{K} \|\mathbf{x}\|_2\right) + \delta, \quad (23)$$

with a tiny safety margin  $\delta \in (0, 10^{-6}]$ . The normalized vector is

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{\gamma}, \quad (24)$$

which satisfies  $K \|\hat{\mathbf{x}}\|_2^2 < 1$ . When mapping back to Euclidean space, we simply undo the scaling:

$$\operatorname{ScaleBack}(\tilde{\mathbf{x}}; \mathbf{x}) = \gamma \tilde{\mathbf{x}}. \quad (25)$$

The exact HypLoRA pipeline is therefore

$$\Delta \mathbf{x} = \gamma \Pi_{H \rightarrow B}^K \left( LLR \left( \Pi_{B \rightarrow H}^K \left( \frac{1}{\gamma} \mathbf{x} \right) \right) \right), \quad (26)$$

where  $LLR$  denotes the Lorentz low-rank transform. Because the same  $\gamma$  is used to contract and expand the vector, the scaling cancels when  $LLR$  is applied to the space-like dimension. This allows us to work with the simplified expression

$$\Delta \mathbf{x} = \Pi_{H \rightarrow B}^K (LLR(\Pi_{B \rightarrow H}^K(\mathbf{x}))), \quad (27)$$

with the understanding that the implementation performs the implicit normalization (23)–(24) and rescaling (25). In practice, these routines only require vector additions, dot products, and scalar divisions, so they are cheap to evaluate and have well-behaved gradients on GPUs. The stereographic route, therefore, keeps Hyperbolic LoRA numerically robust while preserving the geometric signal that motivated the method.

### D.3 Comparisons

In practical implementations for deep learning applications, the simplified stereographic projection offers computational advantages over the exponential and logarithmic maps:

- **Numerical Stability:** The simplified stereographic projection avoids the hyperbolic trigonometric functions ( $\cosh$ ,  $\sinh$ ,  $\tanh$ ) that can lead to numerical overflow for large arguments.
- **Computational Efficiency:** The projection operations involve only basic arithmetic operations, making them computationally efficient compared to the exponential and logarithmic maps.
- **Gradient Flow:** The simplified stereographic projection provides smoother gradient flow during optimization, as it avoids the exponential growth inherent in hyperbolic trigonometric functions.

The simplified stereographic projection thus serves as an alternative parameterization for hyperbolic neural networks, particularly when numerical stability and computational efficiency are primary concerns. This projection can be particularly useful for initializing hyperbolic parameters from Euclidean distributions and for implementing hyperbolic layers that require frequent conversions between Euclidean and hyperbolic representations. The natural correspondence between the Euclidean origin and the hyperbolic origin point  $\mathbf{o}$  facilitates seamless integration with existing neural network architectures.

## E Transformation Analysis

### E.1 Transformation by Exponential and Logarithmic Mappings

*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  $r \ll d$ . Consider the  $n$ -dimensional hyperbolic space  $\mathcal{L}^n$  (Lorentz model) with curvature  $C = -\frac{1}{K}$ , 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 [30], we interpret the Euclidean token embedding  $\mathbf{x}$  as an element in the tangent space at the origin  $\mathbf{o}$  of the hyperbolic space  $\mathcal{L}^n$ . The tangent vector is given by  $\mathbf{v} = (0, \mathbf{x}) \in T_{\mathbf{o}}\mathcal{L}^n$ . The exponential map  $\exp_{\mathbf{o}}^K$  projects  $\mathbf{v}$  onto the hyperbolic space:

$$\exp_{\mathbf{o}}^K(\mathbf{v}) = \left( \sqrt{K} \cosh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right), \sqrt{K} \sinh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right) \frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}} \right), \quad (28)$$

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:

$$\exp_{\mathbf{o}}^K(\mathbf{v}) = \left( \sqrt{K} \cosh\left(\frac{\|\mathbf{x}\|}{\sqrt{K}}\right), \sqrt{K} \sinh\left(\frac{\|\mathbf{x}\|}{\sqrt{K}}\right) \frac{\mathbf{x}}{\|\mathbf{x}\|} \right). \quad (29)$$

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

$$\cosh\left(\frac{z}{\sqrt{K}}\right) \approx 1 + \frac{z^2}{2K}, \quad \sinh\left(\frac{z}{\sqrt{K}}\right) \approx \frac{z}{\sqrt{K}} + \frac{z^3}{6K^{3/2}}. \quad (30)$$

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

$$u_0^H \approx \sqrt{K} + \frac{\|\mathbf{x}\|^2}{2\sqrt{K}}, \quad (31)$$

$$\mathbf{u}_{\text{space}}^H \approx \mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K} \mathbf{x}. \quad (32)$$



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

First transformation:

$$\mathbf{y}_{\text{space}}^H = A\mathbf{u}_{\text{space}}^H \approx A\left(\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}\mathbf{x}\right) = A\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}A\mathbf{x}. \quad (33)$$

Second transformation:

$$\mathbf{z}_{\text{space}}^H = B\mathbf{y}_{\text{space}}^H \approx BA\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \quad (34)$$

Compute the time component after the transformations:

$$z_0^H = \sqrt{K + \|\mathbf{z}_{\text{space}}^H\|^2}. \quad (35)$$

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

$$\Delta Q^{\text{Hyp}} = \log_o^K(\mathbf{z}^H) = \sqrt{K} \cdot \text{arcosh}\left(\frac{z_0^H}{\sqrt{K}}\right) \frac{\mathbf{z}_{\text{space}}^H}{\sqrt{(z_0^H)^2 - K}}. \quad (36)$$

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

$$\text{arcosh}\left(\frac{z_0^H}{\sqrt{K}}\right) \approx \text{arcosh}(1 + \delta) \approx \sqrt{2\delta} = \frac{\|\mathbf{z}_{\text{space}}^H\|}{\sqrt{K}}, \quad (37)$$

$$\sqrt{(z_0^H)^2 - K} \approx \|\mathbf{z}_{\text{space}}^H\|. \quad (38)$$

Therefore, the logarithmic map simplifies to:

$$\Delta Q^{\text{Hyp}} \approx \mathbf{z}_{\text{space}}^H. \quad (39)$$

**Comparing HypLoRA and LoRA Updates.** The HypLoRA update is:

$$\Delta Q^{\text{Hyp}} \approx BA\mathbf{x} + \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \quad (40)$$

The LoRA update is:

$$\Delta Q^{\text{LoRA}} = BA\mathbf{x}. \quad (41)$$

The difference between the updates is:

$$\Delta Q^{\text{Hyp}} - \Delta Q^{\text{LoRA}} = \frac{\|\mathbf{x}\|^2}{6K}BA\mathbf{x}. \quad (42)$$

**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:

$$\text{Scores}_{\text{HypLoRA}} = \frac{(Q^{\text{orig}} + \Delta Q^{\text{Hyp}})(K^{\text{orig}} + \Delta K^{\text{Hyp}})^{\top}}{\sqrt{d_k}}, \quad (43)$$

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

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

$$\Delta \text{Scores} = \text{Scores}_{\text{HypLoRA}} - \text{Scores}_{\text{LoRA}}. \quad (44)$$

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  $C = -\frac{1}{K}$  scales the higher-order corrections. Smaller  $K$  (larger negative curvature) amplifies these terms, aligning with the hyperbolic nature of token embeddings.

**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.

## E.2 Transformation by Stereographic Projection

*Proof.* Let  $\mathbf{x} \in \mathbb{R}^d$ ,  $A \in \mathbb{R}^{r \times d}$ ,  $B \in \mathbb{R}^{d \times r}$  with  $r \ll d$ , and write  $BA : \mathbb{R}^d \rightarrow \mathbb{R}^d$ . We work in the Lorentz model

$$\mathcal{L}_K^d = \{ \mathbf{u} \in \mathbb{R}^{d+1} : \langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{L}} = -K, u_0 > 0 \}, \quad (45)$$

where  $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}} = -u_0 v_0 + \mathbf{u}_{1:d}^\top \mathbf{v}_{1:d}$ , so the sectional curvature is  $-1/K$ . The Poincaré ball is

$$\mathbb{B}_K^d = \left\{ \mathbf{y} \in \mathbb{R}^d : \|\mathbf{y}\| < \frac{1}{\sqrt{K}} \right\}. \quad (46)$$

**Stereographic maps (general  $K$ ).**

$$\Pi_{B \rightarrow H}^K(\mathbf{y}) = \left( \sqrt{K} \frac{1 + K\|\mathbf{y}\|^2}{1 - K\|\mathbf{y}\|^2}, \frac{2\sqrt{K}\mathbf{y}}{1 - K\|\mathbf{y}\|^2} \right), \quad (47)$$

$$\Pi_{H \rightarrow B}^K(\mathbf{u}) = \frac{\mathbf{u}_{1:d}}{u_0 + \sqrt{K}}. \quad (48)$$

These are mutual inverses for  $\|\mathbf{y}\| < 1/\sqrt{K}$ . If needed, softly scale  $\mathbf{x}$  into  $\mathbb{B}_K^d$ ; the scalings cancel in the round trip.

**Pipeline.**

$$\mathbf{x} \xrightarrow{\Pi_{B \rightarrow H}^K} \mathbf{u} \xrightarrow{F} \mathbf{u}' \xrightarrow{\Pi_{H \rightarrow B}^K} \mathbf{x}', \quad \Delta Q^{\text{Stereo}} := \mathbf{x}' - \mathbf{x}. \quad (49)$$

**Space-like action of  $BA$  and renormalization.**

$$\mathbf{u} = (u_0, \mathbf{u}_{1:d}) \mapsto \tilde{\mathbf{u}} = (u_0, (I_d + BA)\mathbf{u}_{1:d}) \mapsto \mathbf{u}' = (u'_0, \mathbf{u}'_{1:d}), \quad (50)$$

$$\mathbf{u}'_{1:d} = (I_d + BA)\mathbf{u}_{1:d}, \quad u'_0 = \sqrt{K + \|\mathbf{u}'_{1:d}\|^2}, \quad (51)$$

which restores  $\langle \mathbf{u}', \mathbf{u}' \rangle_{\mathcal{L}} = -K$ .

**Linearization (first order in  $\|BA\|$ ).** Let  $D := u_0 + \sqrt{K}$  so that  $\mathbf{x} = \Pi_{H \rightarrow B}^K(\mathbf{u}) = \mathbf{u}_{1:d}/D$  and hence  $\mathbf{u}_{1:d} = D\mathbf{x}$ . Put  $\boldsymbol{\delta} := (BA)\mathbf{u}_{1:d} = D(BA)\mathbf{x}$ . Then

$$u'_0 - u_0 = \frac{\mathbf{u}_{1:d}^\top \boldsymbol{\delta}}{u_0} = \frac{D^2}{u_0} (\mathbf{x}^\top (BA)\mathbf{x}) + \mathcal{O}(\|BA\|^2), \quad (52)$$

$$\mathbf{x}' = \frac{\mathbf{u}_{1:d} + \boldsymbol{\delta}}{u_0 + \sqrt{K} + (u'_0 - u_0)} = \frac{\mathbf{u}_{1:d} + \boldsymbol{\delta}}{D} - \frac{\mathbf{u}_{1:d}}{D^2} (u'_0 - u_0) + \mathcal{O}(\|BA\|^2). \quad (53)$$

Using  $\mathbf{u}_{1:d} = D\mathbf{x}$  and the forward map with  $\mathbf{y} = \mathbf{x}$ ,

$$u_0 = \sqrt{K} \frac{1 + K\|\mathbf{x}\|^2}{1 - K\|\mathbf{x}\|^2}, \quad D = u_0 + \sqrt{K} = \frac{2\sqrt{K}}{1 - K\|\mathbf{x}\|^2}, \quad \frac{D}{u_0} = \frac{2}{1 + K\|\mathbf{x}\|^2}. \quad (54)$$

Therefore,

$$\Delta Q^{\text{Stereo}} = (BA)\mathbf{x} - \frac{2}{1 + K\|\mathbf{x}\|^2} (\mathbf{x}^\top (BA)\mathbf{x}) \mathbf{x} + \mathcal{O}(\|BA\|^2). \quad (55)$$

### Comparison with LoRA.

$$\Delta Q^{\text{LoRA}} = (BA) \mathbf{x}, \quad \Delta Q^{\text{Stereo}} - \Delta Q^{\text{LoRA}} = -\frac{2}{1 + K\|\mathbf{x}\|^2} (\mathbf{x}^\top (BA) \mathbf{x}) \mathbf{x} + \mathcal{O}(\|BA\|^2). \quad (56)$$

□

**Remark 2** (Interpretation: self-alignment, norm awareness, and radial balance). *Let  $r := \|\mathbf{x}\|$  and  $g(\mathbf{x}) := \mathbf{x}^\top (BA) \mathbf{x}$ . The stereographic correction is purely radial (aligned with  $\mathbf{x}$ ), with curvature-/norm-dependent gain  $2/(1 + Kr^2)$ :*

$$\Delta Q^{\text{Stereo}} = (BA) \mathbf{x} - \frac{2}{1 + Kr^2} g(\mathbf{x}) \mathbf{x}. \quad (57)$$

Decompose with  $P_{\parallel} = \frac{\mathbf{x}\mathbf{x}^\top}{r^2}$  and  $P_{\perp} = I_d - P_{\parallel}$ :

$$\Delta Q^{\text{Stereo}} = P_{\perp}(BA \mathbf{x}) + \left( \frac{1 + (K - 2)r^2}{1 + Kr^2} \right) g(\mathbf{x}) \frac{\mathbf{x}}{r^2}. \quad (58)$$

Thus the tangential part  $P_{\perp}(BA \mathbf{x})$  is identical to LoRA’s, while the radial part is gated by

$$s_K(r) := \frac{1 + (K - 2)r^2}{1 + Kr^2}. \quad (59)$$

The first-order change of squared norm satisfies

$$\Delta \|\mathbf{x}\|^2 \approx 2 \mathbf{x}^\top \Delta Q^{\text{Stereo}} = 2 \frac{1 + (K - 2)r^2}{1 + Kr^2} g(\mathbf{x}), \quad (60)$$

i.e., LoRA’s  $2g(\mathbf{x})$  is attenuated by  $\frac{1 + (K - 2)r^2}{1 + Kr^2}$ . For fixed  $r$ , smaller  $K$  (more negative curvature  $-1/K$ ) strengthens the correction; for fixed  $K$ , larger  $r$  weakens it as  $r \rightarrow 1/\sqrt{K}$ .

The stereographic transformation implies a geometry-aware, input-adaptive gate on the LoRA update: it preserves the tangential component  $P_{\perp}(BA \mathbf{x})$  while modulating the radial component along  $\mathbf{x}$  by the self-alignment score  $g(\mathbf{x}) = \mathbf{x}^\top (BA) \mathbf{x}$  and the curvature-/norm-dependent factor  $2/(1 + K\|\mathbf{x}\|^2)$ . This indicates that when  $BA \mathbf{x}$  is strongly aligned with  $\mathbf{x}$  (large  $|g(\mathbf{x})|$ ), the method preferentially rebalances the update to reduce excessive norm growth (or amplify radial pull when  $g(\mathbf{x}) < 0$ ), especially for larger-norm embeddings near the ball boundary. Consequently, stereographic favors angular (directional) adjustments over raw radial expansion, leading to more stable representation norms, curvature-aware sensitivity to token specificity (via  $\|\mathbf{x}\|$ ), and a transparent control knob through  $K$  for trading off hierarchical emphasis versus norm preservation.

### E.3 Difference with Hyperbolic Dimension Reduction Methods

Our method does involve hyperbolic dimension reduction. However, HypLoRA is fundamentally distinct from prior hyperbolic dimension reduction techniques such as HoroPCA [98], Principal Geodesic Analysis (PGA) [99], and Exact Principal Geodesic Analysis (EPGA) [100]. These classical methods are primarily designed for *unsupervised* learning: they seek to preserve variance or capture principal directions in hyperbolic space, supporting exploratory data analysis or compact geometric representation.

In contrast, HypLoRA addresses a *supervised, task-specific* goal—parameter-efficient adaptation of LLMs. Rather than maximizing variance without reference to downstream objectives, HypLoRA directly optimizes low-rank hyperbolic adaptations via end-task supervision. This enables scalable and efficient fine-tuning for modern neural architectures.

Furthermore, while both classes of methods operate in hyperbolic geometry, HypLoRA introduces a geometrically meaningful, norm-dependent adaptation through the term  $\|\mathbf{x}\|^2 BA \mathbf{x}$ , derived from hyperbolic operations and regularized via a learnable curvature parameter. This mechanism enables principled, input-dependent adaptation strength, in contrast to dimension reduction methods, which do not control adaptation based on input semantics or hierarchical task structure.

Table 8: Comprehensive comparison of parameter-efficient fine-tuning methods on mathematical reasoning tasks. Results marked with \* are from [93], 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
	<b>HypLoRA (Ours)</b>	<b>79.2</b>	<b>49.3</b>	<b>39.2</b>	<b>20.7</b>	<b>44.6</b>
LLaMA-13B	None	65.5	37.5	32.4	15.0	35.5
	Prefix*	66.8	41.4	31.1	15.7	36.4
	Series*	78.6	50.8	44.0	22.0	47.4
	Parallel*	81.1	55.7	43.3	20.5	48.9
	LoRA*	83.6	54.6	47.5	18.5	50.5
	LoRA†	83.5	54.7	48.5	18.5	51.0
	DoRA	83.0	54.6	OOT	18.9	NA
	<b>HypLoRA (Ours)</b>	<b>83.4</b>	<b>55.8</b>	<b>49.7</b>	<b>21.5</b>	<b>52.5</b>
Gemma-7B	None	76.5	60.4	38.4	25.2	48.3
	LoRA	89.5	74.2	61.4	34.3	65.9
	DoRA	90.7	79.2	68.3	42.5	71.8
	<b>HypLoRA (Ours)</b>	<b>89.5</b>	<b>75.6</b>	<b>68.8</b>	<b>46.5</b>	<b>71.0</b>
LLaMA3-8B	None	79.8	50.0	54.7	21.0	52.1
	LoRA	91.2	81.2	69.8	42.1	73.2
	DoRA	90.3	79.8	73.3	40.9	74.1
	<b>HypLoRA (Ours)</b>	<b>89.1</b>	<b>80.6</b>	<b>72.6</b>	<b>46.9</b>	<b>74.5</b>
Gemma3-4B	LoRA	89.0	80.3	67.7	43.5	71.8
	DoRA	89.5	78.8	68.5	46.1	71.9
	<b>HypLoRA (Ours)</b>	<b>90.8</b>	<b>78.9</b>	<b>67.5</b>	<b>48.4</b>	<b>72.0</b>

## 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. To ensure clarity and conciseness in the main text, we primarily reported results comparing HypLoRA with LoRA across different model architectures and reasoning tasks. Here, we extend our analysis to include additional PEFT baselines such as Prefix tuning, Series and Parallel adapters, and DoRA, 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 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 both standard LoRA and DoRA across multiple model families, with particularly notable improvements on more challenging datasets. For the Gemma-7B model, HypLoRA achieves a micro-averaged accuracy of 71.3%, surpassing LoRA’s 68.6% and DoRA’s 68.0%. Similarly, on LLaMA3-8B, HypLoRA reaches 74.2% compared to LoRA’s 71.9% and DoRA’s 72.5%. The improvements are especially pronounced on the AQuA dataset, which requires complex algebraic reasoning—HypLoRA shows substantial 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

Table 9: Comparison on Commonsense Reasoning Tasks

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
	<b>HypLoRA (Ours)</b>	0.70	<b>74.1</b>	<b>87.6</b>	<b>80.6</b>	<b>94.5</b>	<b>84.7</b>	<b>90.4</b>	<b>81.2</b>	<b>85.2</b>	<b>84.8</b>
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
	<b>HypLoRA (Ours)</b>	1.04	<b>70.0</b>	<b>85.3</b>	<b>79.2</b>	<b>91.5</b>	<b>81.1</b>	<b>89.1</b>	<b>75.9</b>	<b>86.4</b>	<b>82.5</b>
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
	<b>HypLoRA (Ours)</b>	0.71	<b>72.8</b>	<b>89.3</b>	<b>80.4</b>	<b>95.1</b>	<b>84.4</b>	<b>95.5</b>	<b>87.5</b>	<b>90.8</b>	<b>87.0</b>

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 moderate gains 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% on LLaMA3-8B versus 70.8% for LoRA. These results align with our theoretical analysis showing that hyperbolic geometry’s ability to model hierarchical structures becomes more advantageous as problem complexity increases. The fact that HypLoRA maintains superior performance across both older (LLaMA-7B, LLaMA-13B) and newer (LLaMA3-8B, Gemma3-4B) 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.2 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.3 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.

## G Case Study

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

These examples demonstrate how HypLoRA consistently provides more accurate reasoning compared to LoRA across different types of mathematical problems. In Case 1, LoRA miscalculates the sum of

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

Method	LLaMA3-8B		Gemma3-4B	
	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 (stereo)	0.7002	30.12	1.0371	14.61
HypLoRA (exp/log)	0.7099	30.14	1.0518	14.62

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

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

test scores as 380 instead of the correct total of 400, leading to an inaccurate average. Recent research by [101] suggests that LLMs primarily perform pattern matching rather than genuine reasoning. In this scenario, the values 380 and 400 are not well distinguished because they appear close in Euclidean space, particularly in regions distant from the origin, which correspond to lower hierarchical levels. HypLoRA with hyperbolic geometry allows for better separation of such specific numerical tokens, resulting in more precise computational outcomes.

In Case 2, LoRA fails to grasp the true nature of the multi-step temporal reasoning task, incorrectly dividing 240 minutes by 4 days without considering the three-week timeframe. HypLoRA correctly interprets the problem by recognizing that the 240 total minutes must be distributed across three weeks, then allocated to 4 days per week, demonstrating superior semantic understanding and multi-step reasoning capabilities.

In Case 3, LoRA makes an arithmetic error in computing  $11,600 - 9,800 = 1,800$  instead of the correct calculation  $11,600 - 8,800 = 2,800$ , resulting in a wrong conclusion about Gary’s truck capacity. HypLoRA correctly performs the sequential calculations and arrives at the accurate answer. These examples illustrate how the hyperbolic geometry employed by HypLoRA enables better handling of complex reasoning tasks, improving both computational accuracy and logical consistency in mathematical problem-solving scenarios.

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

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

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

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