Enhancing LLM Complex Reasoning Capability through Hyperbolic Geometry

Menglin Yang [1] **Aosong Feng** [1] **Bo Xiong** [2] **Jiahong Liu** [3] **Irwin King** [3] **Rex Ying** [1]

Abstract

In the era of foundation models and large language models (LLMs), Euclidean space is the de facto geometric setting. However, recent studies highlight this choice comes with limitations. We investigate the non-Euclidean characteristics of LLMs on complex reasoning tasks, finding that token embeddings and hidden states exhibit significant degree of hyperbolicity, indicating an underlying hyperbolic structure. To exploit this hyperbolicity, we propose Hyperbolic Low-Rank Adaptation (HoRA), which performs low-rank adaptation fine-tuning on LLMs in hyperbolic space. HoRA operates directly on the hyperbolic manifold, avoiding issues caused by exponential and logarithmic maps when embedding and weight matrices reside in Euclidean space. Experiments show that HoRA obviously improves LLM performance on complex reasoning tasks. Especially the improvement is more obvious, up to 17.30% over Euclidean LoRA on the hard-level AQuA dataset.

1. Introduction

Large language models (LLMs) have shown remarkable capabilities in understanding and generating human-like text (Achiam et al., 2023; Touvron et al., 2023; Gemma Team, 2024; Qin et al., 2023; Shen et al., 2024). However, the default Euclidean geometry used for learning representations may not always be optimal (Linial et al., 1995; Suzuki et al., 2021). Recent studies have shown that latent representations learned by deep neural networks exhibit hyperbolic characteristics, suggesting an underlying tree-like and hierarchical structure (Khrulkov et al., 2020; Bdeir et al., 2023; Cetin et al., 2022). Hyperbolic space, with its negative curvature, is well-suited for modeling hierarchical data, yielding remarkable performance (Nickel & Kiela, 2017; 2018; Ganea et al., 2018a; Khrulkov et al., 2020; Cetin et al., 2022).

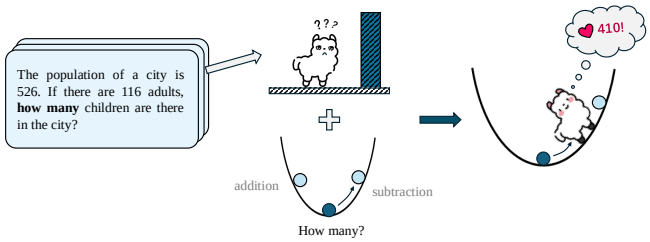


Figure 1. An illustration of how hyperbolic space aids large language models (LLMs) in understanding complex structures in reasoning tasks. In this example, the phrase *How many* is understood as the parent node of *addition* and *subtraction* operations. This hierarchical underlying hierarchical relationship is well captured by hyperbolic space, thereby enhancing the reasoning capabilities of LLMs.

We investigate the hyperbolicity[1] of token embeddings and last hidden states of LLMs on complex reasoning problems. Figure 2 shows the hyperbolicity (δ) distribution generated by LLaMA3-8B[2] on the AQuA dataset (Ling et al., 2017), with additional results provided in Section (2). The consistently low δ values observed across all models suggest a high degree of hyperbolic structure in the representations learned by LLMs. Furthermore, our analysis reveals that the learned representations of complex arithmetic reasoning problems tend to have a larger δ value, or equivalently, a lower degree of hyperbolicity than simpler ones, indicating that the complexity of the reasoning task influences the embedding and hidden state geometry.

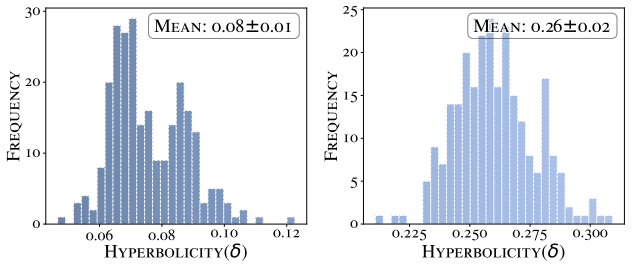
Based on the above findings and the recognized benefits of hyperbolic geometry, a natural consideration is to develop hyperbolic LLMs that explicitly incorporate hyperbolic inductive bias. However, training LLMs from scratch can be resource-intensive (Loshchilov & Hutter, 2017; Rajbhandari et al., 2020), and incorporating Riemannian optimization

[1]Yale University [2]University of Stuttgart [3]The Chinese University of Hong Kong. Correspondence to: Menglin Yang <menglin.yang@outlook.com>.

Workshop of the 41st International Conference on Machine Learning on LLMs and Cognition, Vienna, Austria. PMLR 235, 2024.

[1]Hyperbolicity is a geometric metric that quantifies the deviation of a given metric space from an exact tree metric (Gromov, 1987).

[2]https://ai.meta.com/blog/meta-llama-3/



Metric Space	δ
Sphere Space	0.99±0.01
Dense Graph	0.62±0.01
PubMed Graph	0.40±0.04
Poincare Space	0.14±0.01
Tree Graph	0.0

Figure 2. Hyperbolicity distribution of token embeddings (left) and last hidden states (middle) generated by LLaMA3-8B on the AQuA dataset. The δ value quantifies the degree of hyperbolicity, with values closer to 0 indicating higher hyperbolicity and a more tree-like structure. For better clarity, we also provide δ values in other metric spaces (right), with details in Appendix C.2.

techniques (Kochurov et al., 2020; Smith, 2014; Bécigneul & Ganea, 2018) and additional hyperbolic operations, like Möbius addition (Ganea et al., 2018a; Chami et al., 2019; Chen et al., 2021) could further increase computational demands. As a more resource-efficient alternative, we propose to build a low-rank adaptation method - one of the parameter-efficient fine-tuning approaches (Houlsby et al., 2019; Hu et al., 2021; Zaken et al., 2021; Liu et al., 2022a; Li et al., 2022) in hyperbolic space, named HoRA. Besides, HoRA is particularly advantageous given that existing LLMs are all Euclidean, and not all fine-tuning downstream tasks require hyperbolic geometry. By using hyperbolic adapters for specific tasks on a Euclidean foundation model, we can leverage the benefits of both geometries while maintaining computational efficiency.

Challenge Adapting LLMs in non-Euclidean embedding spaces with existing techniques, i.e., simply applying exponential and logarithmic maps with tangent space for weight adaptation is problematic. This approach fails to fully capture the hyperbolic geometry, as the exponential and logarithmic maps are mutually inverse when applied to representations in the tangent space. 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 Work To address the above challenge, we design HoRA to operate low-rank adaptation directly on the hyperbolic manifold without transformation to the tangent space, thus preserving hyperbolic modeling capabilities and counteracting the reduction. HoRA integrates hyperbolic geometry into existing LLMs, enabling them to benefit from hyperbolic characteristics while minimizing additional computational costs.

Contributions (1) We comprehensively investigate the hyperbolicity of token embeddings and last hidden states in LLMs for complex reasoning problems, revealing their strong hyperbolic properties and the influence of problem complexity on hyperbolicity. (2) We propose HoRA, a parameter-efficient fine-tuning method that integrates hyperbolic geometry into LLMs while preserving hyperbolic modeling capabilities. (3) We demonstrate that HoRA outperforms existing methods, particularly on complex reasoning problems. Our work opens new avenues for exploring the role of geometry in LLMs and provides insights for developing geometrically-informed models for reasoning tasks.

Table 1. Averaged relative hyperbolicity (δ) values for four different datasets. Hyperbolicity is calculated using the Euclidean distance within the original token embedding ('Token') and the last hidden layer state ('Hidden') in LLMs. The accuracy of GPT-3.5 on each dataset is also shown, highlighting that AQuA is the most challenging dataset. L - LLaMA, L3 - LLaMA-3, G - Gemma. 'Token' columns are shaded in gray.

	MAWPS (87.4%)		SVAMP (69.9%)		GSM8K (56.4%)		AQuA (38.9%)	
	Token	Hidden	Token	Hidden	Token	Hidden	Token	Hidden
L-7B	0.08 ± 0.02	0.30 ± 0.03	0.09 ± 0.01	0.32 ± 0.03	0.10 ± 0.01	0.32 ± 0.03	0.10 ± 0.01	0.34 ± 0.02
L-13B	0.08 ± 0.01	0.27 ± 0.03	0.09 ± 0.01	0.28 ± 0.03	0.09 ± 0.01	0.29 ± 0.03	0.10 ± 0.01	0.30 ± 0.02
G-7B	0.11 ± 0.01	0.28 ± 0.05	0.11 ± 0.01	0.27 ± 0.05	0.11 ± 0.01	0.28 ± 0.04	0.12 ± 0.01	0.31 ± 0.04
L3-8B	0.06 ± 0.01	0.23 ± 0.02	0.07 ± 0.01	0.24 ± 0.02	0.07 ± 0.01	0.24 ± 0.02	0.08 ± 0.01	0.26 ± 0.02
Average	0.08 ± 0.01	0.27 ± 0.03	0.09 ± 0.01	0.28 ± 0.03	0.09 ± 0.01	0.28 ± 0.03	0.10 ± 0.01	0.30 ± 0.03

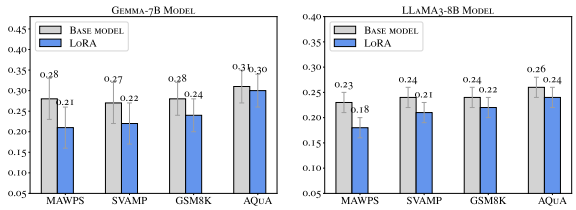


Figure 3. Comparison of Hyperbolicity: Before and after fine-tuning with LoRA.

2. Investigation of Hyperbolicity in LLMs

δ-Hyperbolicity, a concept introduced by Gromov (Gromov, 1987), serves as a measure of the extent to which a metric space (X, d) deviates from an exact tree metric. We consider prompt-level hyperbolicity, where each token in each prompt is treated as a point in a discrete metric space X. This space is spanned by the representations of all tokens within a prompt. To assess the overall hyperbolicity of the LLMs on each dataset, we compute the average hyper-

Table 2. Statistics of datasets

Dataset	Total	#Easy	#Medium	#Hard
MAWPS	238	189	49	0
SVAMP	1,000	691	307	2
GSM8K	1,319	1,098	214	7
AQuA	254	24	178	52
Total	2,811	2,002	748	61

bolicity across all prompts within the respective dataset. We evaluate the hyperbolicity at two levels: the token embedding level and the last hidden layer of the LLMs.

Experimental Settings To investigate the presence of hyperbolicity in large language models (LLMs), we apply the four-point algorithm[3] to various open-source LLMs on arithmetic reasoning datasets, including GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021), where the data details are presented in Table 2. Following the approaches outlined in (Khrulkov et al., 2020) and (Cetin et al., 2022), we estimate the δ-hyperbolicity using the efficient algorithm introduced by Fournier et al. (Fournier et al., 2015). To obtain a scale-invariant measure, we also normalize δ by the diameter of the metric space, $\mathrm{diam}(X)$, resulting in a relative hyperbolicity measure $\delta_{rel} = 2\delta/\mathrm{diam}(X)$ (Borassi et al., 2015). This relative measure ranges from 0 to 1, with values closer to 0 indicating a more hyperbolic, hierarchical, tree-like structure, and values closer to 1 suggesting a perfectly non-hyperbolic space. In Figure 2, we present the hyperbolicity in common metric spaces for reference.

Investigation Results Table 1 presents the average hyperbolicity values (δ) with standard deviations for various LLMs on the respective datasets. The detailed hyperbolicity distribution are presented in Figure 2 and Appendix F. The results reveal several interesting observations:

(1) Both token embeddings and last hidden states exhibit a high degree of hyperbolicity, with the largest δ value in the table being 0.34. This indicates that the representation spaces of LLMs possess a hyperbolic structure, which is a desirable property for capturing the inherent structure of complex reasoning tasks.
(2) Token embeddings consistently demonstrate a higher degree of hyperbolicity compared to the last hidden states across all datasets. This suggests that the initial input representations are highly organized and exhibit strong non-Euclidean patterns, which may be partially lost or transformed during the processing of different tasks by the LLMs.
(3) Smaller degree of hyperbolicity indicates more difficulty of the problem. There is a clear correlation between the difficulty of the dataset and the hyperbolicity values. As the complexity of the dataset increases, as indicated by the lower accuracy scores of GPT-3.5, the corresponding δ values for the last hidden states also increase. This implies that LLMs may struggle to fully capture and maintain the hierarchical structure when dealing with more challenging problems, resulting in lower performance. For instance, AQuA, which is the most challenging dataset with an accuracy of 38.9%, exhibits the highest average δ value of 0.30 for the last hidden states. In contrast, MAWPS, which is the least challenging dataset with an accuracy of 87.4%, has a lower average δ value of 0.27 for the last hidden states. This trend suggests that the ability of LLMs to preserve the hierarchical structure of the input diminishes as the complexity of the reasoning tasks increases.
(4) LoRA fine-tunning will increase the degree of hyperbolicity of learned representation. Figure 3 shows that fine-tuning LLMs with LoRA reduces the δ values across all datasets for both Gemma-7B and LLaMA3-8B models, indicating an increased degree of hyperbolicity in the representation spaces. This suggests that fine-tuning helps LLMs better capture and learn the underlying hierarchical structures in arithmetic reasoning tasks.

[3]The detailed computation of hyperbolicity is given in Appendix C

3. Hyperbolic Low-Rank Adaptation for LLMs

Based on the above investigation, we propose to build hyperbolic low-rank adaptation techniques into LLMs to better capture and preserve the underlying geometries inherent in these representations. The basic idea is illustrated in Figure 1.

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

$$W \otimes \mathbf{x}^H = \exp(W \log_{\mathbf{o}}^K(\mathbf{x}^H)). \tag{1}$$

(Technical Challenge) However, the input from LLMs and the transformation results are in Euclidean space, so we need to apply an additional exponential map and a logarithmic map on the basis of Equation (6) to align the Euclidean representation. This leads to the expression:

$$\begin{aligned}\mathbf{z}^E &= W_{\text{LoRA}}(\mathbf{x}^E) = W\mathbf{x}^E + \Delta W\mathbf{x}^E \\ &= W\mathbf{x}^E + \log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\underbrace{BA\overbrace{\log_{\mathbf{o}}^K(\exp_{\mathbf{o}}^K(\mathbf{x}^E))}^{\mathbf{x}^E}}_{\text{Transformation on } \mathbf{x}^E})) \\ &= W\mathbf{x}^E + BA\mathbf{x}^E,\end{aligned} \tag{2}$$

Table 3. Accuracy comparison of various LLMs with PEFT methods on arithmetic reasoning problems. Results marked with (*) are taken from Hu et al. (Hu et al., 2023). W.AVG. denotes the weighted average accuracy. The results of LoRA on LLaMA-7B/13B reproduced by us are shown in gray. The relative average improvements with respect to LoRA are presented.

Base Model	PEFT	MAWPS	SVAMP	GSM8K	AQuA	W.AVG.
Proportion	NA	8.5%	35.6%	46.9%	9.0%	NA
Hyperbolicity	NA	0.08	0.09	0.09	0.10	NA
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	31.2
	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	44.5
	DoRA	79.0	48.4	39.0	16.4	–
	LoRA	81.9	48.2	38.3	18.5	43.7
	HoRA	79.0	49.1	**39.1**	**20.5** (+10.8%)	**44.3**(+1.4%)
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
	DoRA	**83.8**	55.6	OOT	21.4	–
	LoRA	84.0	54.7	48.5	18.5	51.0
	HoRA	83.2	54.8	**49.0**	**21.7** (+17.3%)	**51.5**(+1.0%)
Gemma-7B	None	76.5	60.4	38.4	25.2	48.3
	DoRA	91.6	75.3	OOT	24.8	–
	LoRA	90.8	77.6	65.6	29.9	68.8
	HoRA	90.3	**79.5**	**66.6**	**32.6** (+9.0%)	**70.1**(+1.8%)
LLaMA3-8B	None	79.8	50.0	54.7	21.0	52.1
	DoRA	**94.5**	80.3	OOT	33.1	–
	LoRA	92.3	79.6	69.7	31.7	71.7
	HoRA	91.6	**81.5**	**71.8**	**34.2** (+7.9%)	**73.5**(+2.5%)

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

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

$$\begin{aligned}\mathbf{z}^E &= W_{\text{LoRA}}(\mathbf{x}^E) = W\mathbf{x}^E + \Delta W\mathbf{x}^E \\ &= W\mathbf{x}^E + \log_{\mathbf{o}}^K(\underbrace{\mathbf{LLR}(BA, \overbrace{\exp_{\mathbf{o}}^K(\mathbf{x}^E)}^{\mathbf{x}^H})}_{\text{Transformation on } \mathbf{x}^H}),\end{aligned} \tag{3}$$

where $\mathbf{LLR}$ represents the direct Lorentz Low-Rank Transformation which is inspired by (Yang et al., 2024; Chen et al., 2021),

$$\mathbf{LLR}(BA, \mathbf{x}^H) = (\sqrt{\|B\mathbf{y}^H\|_2^2 + K}, B\mathbf{y}^H), \tag{4}$$

$$\text{where } \mathbf{y}^H = (\sqrt{\|A\mathbf{x}^H\|_2^2 + K}, A\mathbf{x}^H), \tag{5}$$

where the matrices A and B are still Euclidean parameters, so we do not need to change the original optimizer. It is easy to verify the whole low-rank transformation satisfies the Lorentz constraint, thus preserving the hyperbolic geometry.

Time Complexity. HoRA has similar theoretical time complexity as the Euclidean LoRA, which is $O((d+1)\cdot r + (r+1)\cdot k)$, where d and k represent the input and output dimensions, respectively.

Implementation Details The exponential map scales the original input space with an exponential operator, which is also observed in (Desai et al., 2023). To avoid numerical overflow, before applying the exponential map in Equation (3), we perform L2 normalization on the input and rescale it with a learnable norm scaling factor. The curvature is set as a hyperparameter and searched within the range of $\{0.1, 0.5, 1.0, 2.0\}$. To correctly use the exponential map, following the approach in (Chami et al., 2019), we append a zero to the beginning of the input vector $\mathbf{x}$ to obtain $\mathbf{x}^E$. After applying the logarithmic map, the output vector $\mathbf{z}$ has one additional dimension with zero value. Therefore, we remove this extra dimension from $\mathbf{z}$ to maintain consistency with the original input space.

Table 4. Ablation study

Dataset	Model	Methods		
		LoRA	Tangent	HoRA
AQUA	Gemma-7B	29.9	30.5	**32.6**
	LLaMA3-8B	31.7	32.0	**34.2**
SVAMP	Gemma-7B	77.6	78.2	**79.5**
	LLaMA3-8B	79.6	80.1	**81.5**

3.1. Experimental Results

Table 2 presents the detailed statistics of these test datasets. The "Ratio" represents the proportion of each dataset in the total number of questions across all four datasets.

The experimental setting closely follows the approach in (Hu et al., 2023). The training set for fine-tuning is collected from GSM8K (Cobbe et al., 2021), MAWPS, MAWPS-single (Koncel-Kedziorski et al., 2016), and consists of 1,000 examples. To augment the reasoning capabilities, step-by-step rationales produced by ChatGPT are also included for the training samples, as in (Hu et al., 2023). As a result, a set of 10K math reasoning samples is obtained for training. The test datasets include GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). Although GSM8K and MAWPS also appear in the training set, there is no overlap between the training and test sets. Table 3, Table 4 and Figure 4 present our main experimental results. We have the following findings:

(1) Dataset Difficulty and Model Performance The performance of the models is strongly related to the difficulty level of the datasets. This holds true for both the base models and the fine-tuned results. It is as complex problems require more complex reasoning and a better understanding of the underlying structure of the problem.

(2) Overall Performance of HoRA HoRA consistently outperforms other Parameter-Efficient Fine-Tuning (PEFT) methods across a range of base models and datasets, particularly excelling in the more challenging arithmetic reasoning problems. Notably, HoRA achieves the highest accuracy

improvements on the AQuA dataset across all models, with increases of up to 17.30% compared to LoRA. It also shows robust performance improvements in the GSM8K dataset, further demonstrating its effectiveness in enhancing model reasoning capabilities under complex problem-solving scenarios.

(3) Performance with DoRA model and on MAWPS Dataset Due to DoRA's high time complexity, we were unable to obtain an evaluation value for the DoRA model on GSM8K within the limited timeframe. However, it is important to note that our method is orthogonal to this approach and can be applied independently. Therefore, we mainly compare with LoRA and show the effectiveness of hyperbolic geometry. On the MAWPS dataset, HoRA's performance improvement is less significant compared to other datasets, and in some cases, it is even lower than the baseline PEFT. This can be attributed to the fact that our model uses the same curvature for all datasets finetuning, whereas simpler problems may not require the same level of curvature as more complex ones. To address this issue, we plan to explore adaptive curvature techniques that can adjust to the complexity of individual questions in future work. Despite this limitation, our method has still achieved notable improvements compared to the base model.

(4) Ablation Study We conducted an ablation study using the tangent-space method (Equation (2)) as a baseline to compare with the proposed HoRA. The key difference lies in the approach for the Low-rank Transformation. This comparison helps evaluate the direct Lorentz Low-rank approach's effectiveness. Both Equation (2) and HoRA include an additional rescaling operation (Section 3). The tangent-space method can be seen as vanilla LoRA combined with rescaling. This allows us to assess the impact of normal rescaling. Table 4 shows our results. The tangent-space method improves over the original LoRA due to the flexibility introduced by the rescaling step, which is necessary in hyperbolic geometry. Comparing HoRA with the tangent-space method highlights the significant improvements from incorporating hyperbolic geometry.

(5) The Effectiveness of Ranks The rank of adaptation matrices A and B in HoRA is crucial for expressiveness and efficiency. We experimented with various ranks using the LLaMA3-8B model on the AQuA dataset. Figure 4 shows that lower ranks generally reduce accuracy. However, the performance gap between HoRA and LoRA increases as rank decreases, demonstrating hyperbolic space's effectiveness.

(6) The Effectiveness of Curvatures Curvature in hyperbolic space is a critical hyperparameter in HoRA, affecting its ability to capture underlying structures. We evaluated different curvatures using the LLaMA3-8B model on the AQuA dataset. Results in Figure 4 (right) show that $K = 0.1$ (curvature = -10) excels on hard problems but underperforms on easier ones. Increasing K (e.g., K=1.0) improves performance on easier problems but reduces it on hard ones. This suggests that larger curvatures better handle complex structures, while smaller curvatures perform well on simpler tasks. Performance decreases when the curvature becomes too small, indicating an optimal range for K.

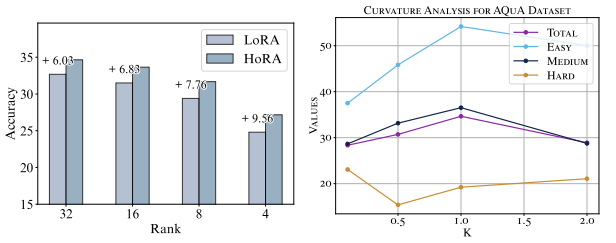


Figure 4. Rank and curvature ($-1/K$) analysis

4. Conclusion

In this study, we investigated the non-Euclidean characteristics of token embeddings and hidden states in LLMs on complex reasoning tasks. Our findings revealed a high degree of hyperbolicity in the representation spaces of LLMs, with more complex problems exhibiting lower hyperbolicity. Building on these insights, we proposed HoRA, a hyperbolic low-rank adaptation method that integrates hyperbolic geometry into the fine-tuning process of LLMs. Through extensive experiments, we demonstrated the effectiveness of HoRA in improving the performance of LLMs on arithmetic reasoning tasks, particularly on medium and hard-level datasets. By leveraging the inherent hyperbolic structure of the data, HoRA enables LLMs to better capture and utilize the complex relationships present in the problems, leading to enhanced reasoning capabilities.

Limitation The understanding of why embeddings in LLMs exhibit hyperbolic properties is still limited, and HoRA uses a fixed curvature for all datasets, which may not be optimal for simpler and harder problems at the same time. In future work, we will explore the theoretical foundations of this phenomenon, investigate adaptive curvature techniques, and provide a more robust framework for integrating hyperbolic geometry into LLMs to enhance the generalizability of our approach.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language

model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Bdeir, A., Schwethelm, K., and Landwehr, N. Hyperbolic geometry in computer vision: A novel framework for convolutional neural networks. *arXiv preprint arXiv:2303.15919*, 2023.

Bécigneul, G. and Ganea, O.-E. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

Borassi, M., Chessa, A., and Caldarelli, G. Hyperbolicity measures democracy in real-world networks. *Physical Review E*, 92(3):032812, 2015.

Cetin, E., Chamberlain, B., Bronstein, M., and Hunt, J. J. Hyperbolic deep reinforcement learning. *arXiv preprint arXiv:2210.01542*, 2022.

Chami, I., Ying, Z., Ré, C., and Leskovec, J. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Chen, W., Han, X., Lin, Y., Zhao, H., Liu, Z., Li, P., Sun, M., and Zhou, J. Fully hyperbolic neural networks. *arXiv preprint arXiv:2105.14686*, 2021.

Chen, Y., Yang, M., Zhang, Y., Zhao, M., Meng, Z., Hao, J., and King, I. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pp. 94–102, 2022.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Desai, K., Nickel, M., Rajpurohit, T., Johnson, J., and Vedantam, R. Hyperbolic Image-Text Representations. In *Proceedings of the International Conference on Machine Learning*, 2023.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Edalati, A., Tahaei, M., Kobyzev, I., Nia, V. P., Clark, J. J., and Rezagholizadeh, M. Krona: Parameter efficient tuning with kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022.

Foundation, O. Introducing chatgpt. https://openai.com/index/chatgpt, November 2022.

Foundation, O. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Fournier, H., Ismail, A., and Vigneron, A. Computing the gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015.

Ganea, O., Bécigneul, G., and Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pp. 1646–1655. PMLR, 2018a.

Ganea, O., Bécigneul, G., and Hofmann, T. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018b.

Gemma Team, G. D. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Gromov, M. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.

Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K. M., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11–15. Pasadena, CA USA, 2008.

Hayou, S., Ghosh, N., and Yu, B. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E.-P., Lee, R. K.-W., Bing, L., and Poria, S. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

Key, O., Kaddour, J., and Minervini, P. Local lora: Memory-efficient fine-tuning of large language models. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization*, 2023.

Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., and Lempitsky, V. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6418–6428, 2020.

Kochurov, M., Karimov, R., and Kozlukov, S. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*, 2020.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1152–1157, 2016.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Li, S., Lu, H., Wu, T., Yu, M., Weng, Q., Chen, X., Shan, Y., Yuan, B., and Wang, W. Caraserve: Cpu-assisted and rank-aware lora serving for generative llm inference. *arXiv preprint arXiv:2401.11240*, 2024.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., and Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Linial, N., London, E., and Rabinovich, Y. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965, 2022a.

Liu, J., Yang, M., Zhou, M., Feng, S., and Fournier-Viger, P. Enhancing hyperbolic graph embeddings via contrastive learning. *arXiv preprint arXiv:2201.08554*, 2022b.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Mettes, P., Atigh, M. G., Keller-Ressel, M., Gu, J., and Yeung, S. Hyperbolic deep learning in computer vision: A survey. *arXiv preprint arXiv:2305.06611*, 2023.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pp. 6338–6347, 2017.

Nickel, M. and Kiela, D. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pp. 3779–3788, 2018.

Patel, A., Bhattamishra, S., and Goyal, N. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

Peng, W., Varanka, T., Mostafa, A., Shi, H., and Zhao, G. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., and Yang, D. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

Qin, Y., Wang, X., Su, Y., Lin, Y., Ding, N., Yi, J., Chen, W., Liu, Z., Li, J., Hou, L., et al. Exploring universal intrinsic task subspace via prompt tuning. *arXiv preprint arXiv:2110.07867*, 2021.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Ren, P., Shi, C., Wu, S., Zhang, M., Ren, Z., de Rijke, M., Chen, Z., and Pei, J. Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.17263*, 2024.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Shen, Y., Song, K., Tan, X., Li, D., Lu, W., and Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. *Advances in Neural Information Processing Systems*, 36, 2024.

Sheng, Y., Cao, S., Li, D., Hooper, C., Lee, N., Yang, S., Chou, C., Zhu, B., Zheng, L., Keutzer, K., et al. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*, 2023.

Shimizu, R., Mukuta, Y., and Harada, T. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

Smith, S. T. Optimization techniques on riemannian manifolds. *arXiv preprint arXiv:1407.5965*, 2014.

Sun, J., Cheng, Z., Zuberi, S., Pérez, F., and Volkovs, M. HGCF: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference*, pp. 593–601, 2021.

Suzuki, A., Nitanda, A., Wang, J., Xu, L., Yamanishi, K., and Cavazza, M. Generalization error bound for hyperbolic ordinal embedding. In *International Conference on Machine Learning*, pp. 10011–10021. PMLR, 2021.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Van Spengler, M., Berkhout, E., and Mettes, P. Poincaré ResNet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5419–5428, 2023.

Wang, X., Aitchison, L., and Rudolph, M. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

Weng, Z., Ogut, M. G., Limonchik, S., and Yeung, S. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2603–2612, 2021.

Xiong, B., Cochez, M., Nayyeri, M., and Staab, S. Hyperbolic embedding inference for structured multi-label prediction. *Advances in Neural Information Processing Systems*, 35:33016–33028, 2022.

Xu, Y., Xie, L., Gu, X., Chen, X., Chang, H., Zhang, H., Chen, Z., Zhang, X., and Tian, Q. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

Yang, M., Zhou, M., Kalander, M., Huang, Z., and King, I. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1975–1985, 2021.

Yang, M., Li, Z., Zhou, M., Liu, J., and King, I. HICF: Hyperbolic informative collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2212–2221, 2022a.

Yang, M., Zhou, M., Li, Z., Liu, J., Pan, L., Xiong, H., and King, I. Hyperbolic graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852*, 2022b.

Yang, M., Zhou, M., Liu, J., Lian, D., and King, I. HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization. In *Proceedings of the ACM Web Conference 2022*, pp. 2462–2471, 2022c.

Yang, M., Zhou, M., Xiong, H., and King, I. Hyperbolic temporal network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2022d.

Yang, M., Zhou, M., Pan, L., and King, I. κHGCN: Tree-likeness modeling via continuous and discrete curvature learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2965–2977, 2023a.

Yang, M., Zhou, M., Ying, R., Chen, Y., and King, I. Hyperbolic representation learning: Revisiting and advancing. In *International Conference on Machine Learning*, pp. 39639–39659. PMLR, 2023b.

Yang, M., Verma, H., Zhang, D. C., Liu, J., King, I., and Ying, R. Hypformer: Exploring efficient hyperbolic transformer fully in hyperbolic space. *arXiv preprint arXiv:2407.01290*, 2024.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Zeng, Y. and Lee, K. The expressive power of low-rank adaptation. *arXiv preprint arXiv:2310.17513*, 2023.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhu, J., Greenewald, K., Nadjahi, K., Borde, H. S. d. O., Gabrielsson, R. B., Choshen, L., Ghassemi, M., Yurochkin, M., and Solomon, J. Asymmetry in low-rank adapters of foundation models. *arXiv preprint arXiv:2402.16842*, 2024.

Zhu, Y., Feng, J., Zhao, C., Wang, M., and Li, L. Counter-interference adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 2021.

Appendix

A. Related Work

Parameter Efficient Fine Tuning (PEFT) and LoRAs Fine-tuning LLMs (Foundation, 2022; 2023; Touvron et al., 2023) for downstream tasks poses significant challenges due to their massive number of parameters, often reaching billions or even trillions. To address this issue, PEFT methods have been proposed, which aim to train a small subset of parameters while achieving better performance compared to full fine-tuning. PEFT methods can be broadly categorized into prompt-based methods (Lester et al., 2021; Li & Liang, 2021; Qin et al., 2021), adapter-based methods (Houlsby et al., 2019; Zhu et al., 2021),and reparameterization-based methods (Hu et al., 2021; Aghajanyan et al., 2020; Edalati et al., 2022). Among these, LoRA (Hu et al., 2021) as the reparameterization-based method, has gained significant attention due to its simplicity, effectiveness, and compatibility with existing model architectures. Variants of LoRA, such as LoRA+(Hayou et al., 2024), DoRA (Liu et al., 2024) and AdaLoRA (Zhang et al., 2023), have been proposed to improve its performance and efficiency. Recent research has also investigated the deployment of LoRA in resource-constrained environments (Sheng et al., 2023; Li et al., 2024; Key et al., 2023), ensembles of multiple LoRAs (Wang et al., 2023; Ren et al., 2024), quantization techniques (Dettmers et al., 2024; Xu et al., 2023; Li et al., 2023), and theoretical properties (Zeng & Lee, 2023; Zhu et al., 2024). Despite these advances, existing methods operate within Euclidean space, ignoring the underlying structure represented by LLMs. Our approach, as an orthogonal method, addresses this gap by integrating hyperbolic geometry into LoRA.

Hyperbolic Representation Learning and Deep Learning Hyperbolic geometry has been successfully applied to various neural network architectures and models (Yang et al., 2022b; Mettes et al., 2023; Peng et al., 2021), including shallow hyperbolic neural networks (Ganea et al., 2018a;b; Chen et al., 2021; Shimizu et al., 2020), hyperbolic CNNs (Bdeir et al., 2023; Van Spengler et al., 2023), and hyperbolic attention networks or Transformers (Gulcehre et al., 2018; Chen et al., 2021; Shimizu et al., 2020). These models leverage the inductive biases of hyperbolic geometry to achieve remarkable performance on various tasks and applications (Chami et al., 2019; Yang et al., 2022a; Sun et al., 2021; Khrulkov et al., 2020; Cetin et al., 2022; Weng et al., 2021; Xiong et al., 2022; Yang et al., 2021; 2022c;d; Liu et al., 2022b; Chen et al., 2022; Yang et al., 2023a;b). However, training LLMs from scratch remains computationally expensive (Kochurov et al., 2020; Smith, 2014). To address this challenge and enable the efficient integration of hyperbolic geometry into LLMs, we propose HoRA. This approach leverages the strengths of both hyperbolic geometry and LoRA to achieve efficient and effective model training.

B. Preliminary

This section introduces key concepts: LoRA adapter, Lorentz model of hyperbolic geometry, hyperbolic linear transformations, and hyperbolicity.

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

$$\mathbf{z} = W_{\text{LoRA}}(\mathbf{x}) = W\mathbf{x} + \Delta W\mathbf{x} = W\mathbf{x} + BA\mathbf{x}. \tag{6}$$

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

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

model in n dimensions with curvature $-1/K(K > 0)$ is defined as:

$$\mathcal{L}_K^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -K, x_0 > 0\}, \tag{7}$$

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

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

$$\mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n := \{\mathbf{u} \in \mathbb{R}^{n+1} : \langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}} = 0\}. \tag{8}$$

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

Consider a point $\mathbf{x} \in \mathcal{L}_K^n$ and a tangent vector $\mathbf{u} \in \mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n$. The exponential map, denoted as $\exp_{\mathbf{x}}^K : \mathcal{T}_{\mathbf{x}}\mathcal{L}_K^n \to \mathcal{L}_K^n$, assigns to $\mathbf{u}$ the point $\exp_{\mathbf{x}}^K(\mathbf{u}) := \gamma(1)$, where γ represents the unique geodesic that satisfies the initial conditions $\gamma(0) = \mathbf{x}$ and $\dot{\gamma}(0) = \mathbf{u}$.

The **exponential map** can be explicitly expressed as follows:

$$\exp_{\mathbf{x}}^K(\mathbf{u}) = \cosh\left(\frac{\|\mathbf{u}\|_{\mathcal{L}}}{\sqrt{K}}\right)\mathbf{x} + \sqrt{K}\sinh\left(\frac{\|\mathbf{u}\|_{\mathcal{L}}}{\sqrt{K}}\right)\frac{\mathbf{u}}{\|\mathbf{u}\|_{\mathcal{L}}}, \tag{9}$$

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

The **logarithmic map** $\log_{\mathbf{u}}^K(\mathbf{x}) : \mathcal{L}_K^n \to \mathcal{T}_{\mathbf{u}}\mathcal{L}_K^n$ plays an inverse role. It is defined by the equation:

$$\log_{\mathbf{u}}^K(\mathbf{x}) = \frac{\cosh^{-1}\left(-\frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}}\right)}{\sinh\left(\cosh^{-1}\left(-\frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}}\right)\right)}\left(\mathbf{x} + \frac{1}{K}\langle \mathbf{u}, \mathbf{x} \rangle_{\mathcal{L}}\mathbf{u}\right). \tag{10}$$

The exponential and logarithmic maps establish a bijective projection between the tangent space and hyperbolic space. Notably, $\log_{\mathbf{x}}^K(\exp_{\mathbf{x}}^K(\mathbf{u})) = \mathbf{u}$ and $\exp_{\mathbf{u}}^K(\log_{\mathbf{u}}^K(\mathbf{x})) = \mathbf{x}$. Consequently, Equation (2) 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 $\mathbf{o} := \{\sqrt{K}, 0, \ldots, 0\} \in \mathcal{L}_K^n$ as a common reference point.

C. δ-Hyperbolicity

δ-Hyperbolicity, a concept introduced by Gromov (Gromov, 1987), serves as a measure of the extent to which a metric space (X, d) deviates from an exact tree metric. This concept is particularly relevant when investigating the hierarchical properties of representation spaces learned by LLMs. A lower δ-hyperbolicity value, or equivalently, a higher degree of hyperbolicity, indicates a more tree-like structure within the space. The δ-hyperbolicity of a space can be quantified using the four-point condition.

C.1. δ-Hyperbolicity Computation

For any four points a, b, c, and w in the space, the Gromov product $[a, c]_w$ at point w is bounded below by the minimum of the Gromov products $[a, b]_w$ and $[b, c]_w$, minus a slack term δ:

$$[a, c]_w \geq \min([a, b]_w, [b, c]_w) - \delta, \tag{11}$$

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

$$[a, b]_w = \frac{1}{2}(d(a, w) + d(b, w) - d(a, b)). \tag{12}$$

A metric space X is considered δ-hyperbolic if the four-point condition holds for all points a, b, c, and w in the space. In geodesic metric spaces, δ-hyperbolicity implies that geodesic triangles are δ-slim. This means that for any point on one side of a triangle, there exists a point on one of the other sides within a distance of δ. In the case of an exact tree metric, where the sides of any triangle intersect at a single point, the δ-hyperbolicity value is zero. This is because the four-point condition is satisfied with equality for all points in the space.

Table 5. Sample Questions. Easy, Medium, and Hard denote the difficulty level.

Easy	The population of a city is 5,265,526. If there are 4,169,516 adults in the city, how many children are there in the city?
Medium	If $6x - y = 24$ and $y = 3x$, what is the value of x?
Hard	A rectangular solid, 3 x 4 x 15, is inscribed in a sphere so that all eight of its vertices are on the sphere. What is the diameter of the sphere?

Table 6. The average relative hyperbolicity values (δ) were calculated for various difficulty levels. These averages were computed across four datasets. The 'Hard' row are in gray.

	LLaMA-7B		LLaMA-13B		LLaMA3-8B		Gemma-7B	
	Token	Hidden	Token	Hidden	Token	Hidden	Token	Hidden
Easy	0.09 ± 0.01	0.31 ± 0.03	0.09 ± 0.01	0.28 ± 0.03	0.07 ± 0.01	0.23 ± 0.02	0.11 ± 0.01	0.27 ± 0.05
Medium	0.10 ± 0.01	0.32 ± 0.03	0.09 ± 0.01	0.29 ± 0.03	0.07 ± 0.01	0.24 ± 0.02	0.11 ± 0.01	0.29 ± 0.04
Hard	0.11 ± 0.02	0.33 ± 0.03	0.10 ± 0.02	0.30 ± 0.02	0.08 ± 0.01	0.26 ± 0.02	0.12 ± 0.01	0.31 ± 0.03

C.2. Hyperbolicity on Different Metric Spaces

Figure 2 presents the hyperbolicity values in both continuous (i.e., Poincaré Space and Sphere Space) and discrete metric spaces (i.e., Random Tree Graph, Dense Graph and PubMed Graph). We employ a consistent processing method, akin to the one mentioned in the Section (2) for embedding spaces. Specifically, we sample 1000 4-tuples, compute the delta value for each, and then take the maximum value.

For the Poincaré ball and sphere spaces, we use a two-dimensional model and calculate hyperbolicity based on their respective geodesic distances. The PubMed graph is sourced from Sen et al. (Sen et al., 2008). The tree graph and dense graph are generated using NetworkX (Hagberg et al., 2008). For these graphs, we first remove isolated nodes before performing our calculations in a consistent manner. The shortest path distance on the graph is used as the distance measure, analogous to the concept of geodesics in continuous spaces.

D. Difficulty Level and Hyperbolicity

In this study, we further utilize the GPT4-turbo API and manual review evaluation to label the difficulty level for each test dataset, categorizing them into easy, medium, and hard levels. Easy-level questions are simple and can be solved in a single step using basic arithmetic. Medium-level questions require multiple steps. Hard-level questions are advanced, requiring complex reasoning. The detailed instructions are given in the Appendix E. Table 2 presents the detailed statistics of these test datasets. The "Category" column denotes the question type, and the "Ratio" represents the proportion of each dataset in the total number of questions across all four datasets. Table 5 further provides examples of easy, medium, and hard-level questions for better understanding.

For the baseline model comparison, the LLaMA-7B and 13B base models mentioned in (Hu et al., 2023) are included, as well as Prefix-Tuning (Li & Liang, 2021), Series Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021), and Parallel Adapter (He et al., 2021). Additionally, the recently released base models Gemma-7B and LLaMA3-8B are fine-tuned using LoRA for comparison. It is worth noting that the final results are not directly averaged, as the four datasets contain different numbers of questions, for instance, 1,319 in GSM8K and 238 in MAWPS. Instead, a weighted average is performed based on the number of questions in each dataset to ensure a fair comparison. For the LoRA implementation, it is inserted simultaneously into both the Multi-head Attention layers and MLP layers in the base model. All experiments are run on a single GPU: NVIDIA A40 or A100.

From the above labeling, it can be observed that at the dataset level, MAWPS is an easy-level dataset, SVAMP and GSM8K are medium-level datasets, and AQuA is a hard-level dataset. This observation aligns with the reported hyperbolicity in Table 1 and the accuracy of ChatGPT's answers, further verifying the third point discovered in the experiments in Section 2: a larger hyperbolicity value, or equivalently, a smaller degree of hyperbolicity indicates a more difficult problem.

E. Instructions for Generating Difficulty Levels

To systematically assign a difficulty level to each question in the dataset, we employed a rigorous labeling method using the GPT-4 Turbo API. Each question was evaluated three times to ensure consistency, and any discrepancies were resolved through manual review.

> **Global Prompt**: You will rate math problems based on their complexity. Level 0 is simple and can be solved in a single step using basic arithmetic. Level 1 is intermediate, requiring multiple steps. Level 2 is advanced, requiring complex reasoning.
>
> **Per Question Prompt**: Please analyze the complexity of the following math problem based on the earlier criteria. Here is the problem:
>
> Output format: Output the numbers 0, 1, or 2.

The Global Prompt is provided at the beginning, while the Per Question Prompt is used for each question in the dataset.

To further investigate the relationship between problem complexity and hyperbolicity, we grouped the examples from the dataset according to their difficulty levels: easy, medium, and hard. We then calculated the hyperbolicity values for each difficulty level, and the results are presented in Table 6. The data in Table 6 reveals a strong correlation between the complexity of the problems and their corresponding hyperbolicity values. Notably, hard-level problems exhibit lower degrees of hyperbolicity compared to simple arithmetic problems. This finding suggests that LLMs can more readily discern the hierarchical structure of simple problems, whereas they encounter challenges when dealing with hard-level problems.

This observation is further validated by the accuracy of LLMs' predictions across different difficulty levels, with lower accuracy rates observed for more complex problems (e.g., AQuA). The relationship between problem complexity and hyperbolicity highlight the importance of considering the underlying geometry when evaluating and developing LLMs for mathematical problem-solving tasks. This is also the problem that this work is to solve.

F. Comprehensive Hyperbolicity Distribution

In Figure 2, we display the hyperbolicity distribution of LLaMA-8B on the AQuA dataset due to space limitations. Here, we present all observed hyperbolicity distributions. The mean and variance of these distributions are recorded in Table 1. Additionally, we use a consistent colormap, where dark blue indicates lower values of hyperbolicity and lighter colors indicate higher values of hyperbolicity.

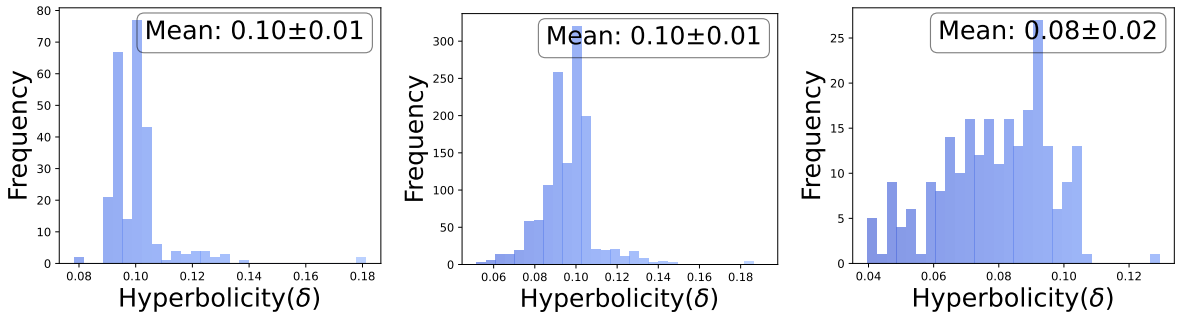


Figure 5. Hyperbolicity Distribution of Token Embedding by LLaMA-7B. Datasets: AQuA (left), GSM8K (middle), MAWPS (right).

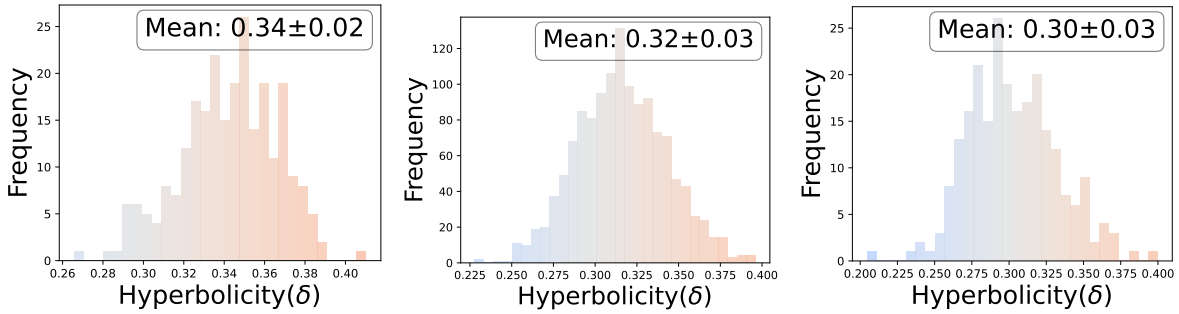


Figure 6. Hyperbolicity Distribution of Last Hidden Layer by LLaMA-7B. Datasets: AQuA (left), GSM8K (middle), MAWPS (right).

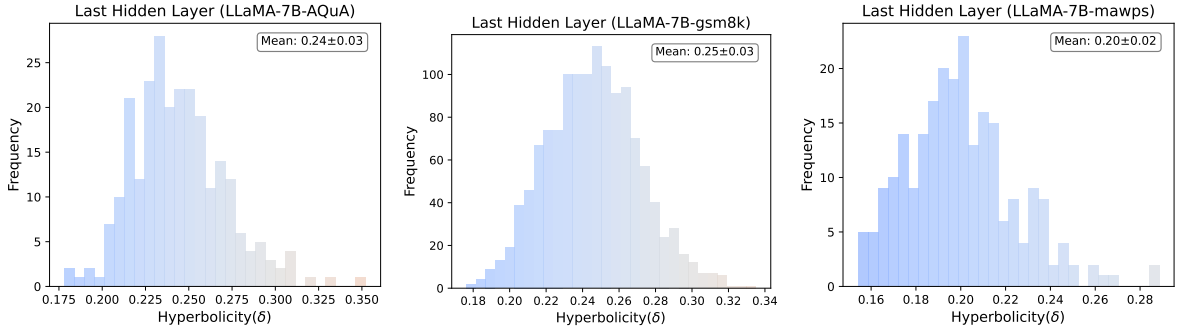


Figure 7. Hyperbolicity Distribution of Last Hidden Layer by LLaMA-7B with LoRA. Datasets: AQuA (left), GSM8K (middle), MAWPS (right).

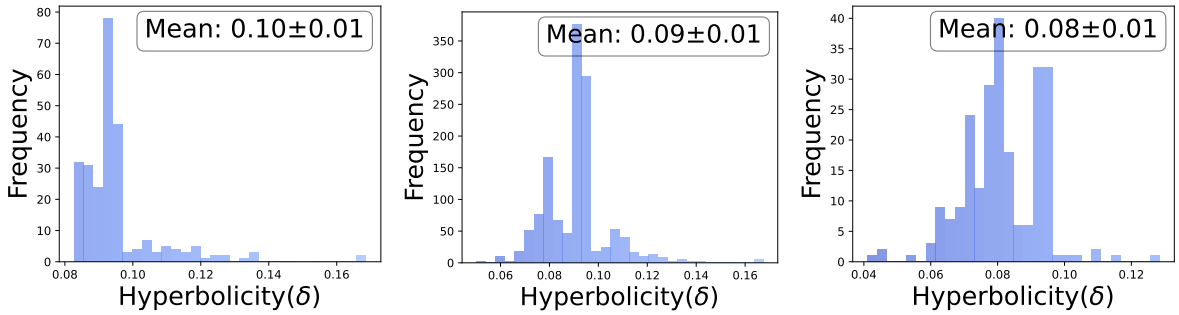


Figure 8. Hyperbolicity Distribution of Token Embedding by LLaMA2-13B. Data sets: AQuA (left), GSM8K (middle), MAWPS (right).

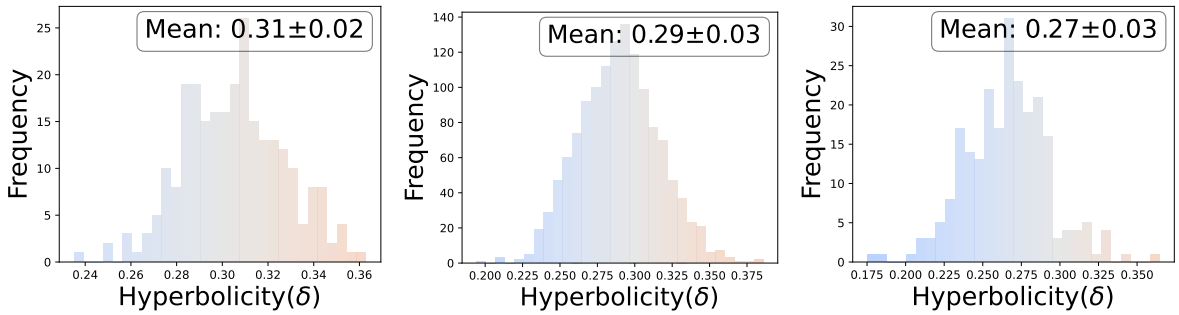


Figure 9. Hyperbolicity Distribution of Last Hidden Layer by LLaMA2-13B. Datasets: AQuA (left), GSM8K (middle), MAWPS (right).

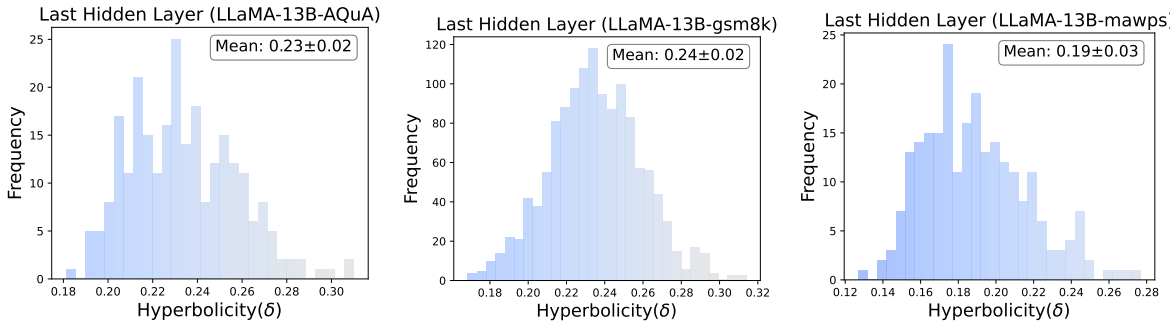


Figure 10. Hyperbolicity Distribution of Last Hidden Layer by LLaMA-13B with LoRA. Datasets: AQuA (left), GSM8K (middle), MAWPS (right).