PARAMETER-EFFICIENT FINE-TUNING OF LLMS WITH MIXTURE OF SPACE EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have achieved remarkable progress, with Parameter-Efficient Fine-Tuning (PEFT) emerging as a key technique for downstream task adaptation. However, existing PEFT methods mainly operate in Euclidean space, fundamentally limiting their capacity to capture complex geometric structures inherent in language data. Although hyperbolic and spherical geometries provide useful inductive biases for hierarchical and circular structures, respectively, imposing a single manifold type still limits representational expressiveness, even with learnable curvature. To address this, we propose Mixture of Space (MoS), a unified framework that leverages multiple geometric spaces simultaneously to learn richer, curvature-aware representations. Building on this scheme, we develop MoSLoRA, which extends Low-Rank Adaptation (LoRA) with heterogeneous geometric experts, enabling models to dynamically select or combine appropriate geometric spaces based on input context. Furthermore, MoSLoRA integrates lightweight token routing to select suitable geometric experts for each input token and aggregates their outputs through a unified projection, avoiding frequent exponential and logarithmic manifold mappings while preserving multi-space specialization. Our experiments across diverse benchmarks demonstrate that MoSLoRA outperforms strong baselines, achieving up to 5.6% improvement on MATH500 and 15.9% on MAWPS. Moreover, we provide empirical analyses of curvature dynamics, revealing how adaptive space selection and curvature optimization affect training stability and model performance. The code and datasets are available at https://anonymous.4open.science/r/MoSLoRA.

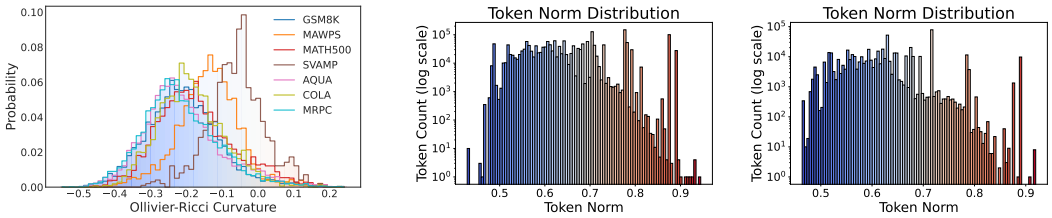
1 INTRODUCTION

Large Language Models (LLMs) have recently demonstrated impressive performance across a wide range of applications, including translation, comprehension, dialogue, and reasoning (Achiam et al., 2023; Jaech et al., 2024; Dubey et al., 2024; Yang et al., 2024a). With the aid of post-training techniques such as instruction tuning, they can be further adapted to diverse downstream tasks with notable gains in effectiveness (Hu et al., 2023; Han et al., 2024). Despite these advances, most existing approaches rely on a Euclidean assumption, modeling all embeddings in flat Euclidean space, which are often inadequate to capture the semantic diversity and contextual complexity of natural language (Bronstein et al., 2017; Park et al., 2024; He et al., 2025a).

Semantic structures in language often display geometric patterns: hierarchical relationships where broad and general concepts naturally encompass finer subcategories and distinct entities; circular patterns among synonymous expressions or co-referential terms (such as days of a week and months of a year) (Engels et al., 2024); and complex multi-level dependencies that resist simple linear organization. These patterns are largely overlooked and constrained under Euclidean representations, leaving open the question of *how to effectively leverage such naturally occurring structures within embedding spaces to unlock richer representational capacity?*

Recently, growing attention has shifted toward non-Euclidean representation spaces for improving language model adaptation and reasoning (Peng et al., 2021; Yang et al., 2024c; Pal et al., 2024; Loshchilov et al., 2024). These works suggest that different geometries provide useful inductive biases for different structural patterns: hyperbolic space is well suited for hierarchical relations, while spherical space can naturally encode circular structures. HypLoRA introduces negative-curvature

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



(a) Corpus-level Ricci curvature.

(b) Token embedding norm distributions on MATH and QA datasets.

Figure 1: Empirical evidence of heterogeneous non-Euclidean geometry in token representations. (a) Corpus-level Ollivier–Ricci curvature distributions, where the x-axis is the estimated curvature and the y-axis is the probability. (b) Token norm distributions on MATH(left) and QA(right) datasets. The coexistence of dominant negative curvature, non-negligible positive curvature, and long-tailed norm distributions suggests that token representations cannot be fully characterized by a single geometry.

geometry into parameter-efficient fine-tuning to reduce embedding distortion (Yang et al., 2024b), whereas nGPT reformulates Transformer representations on hyperspheres through normalization constraints (Loshchilov et al., 2024). However, these approaches are designed around a particular geometric assumption, leaving open how to accommodate heterogeneous local structures within the same task. To examine whether various geometric structures arise in task-relevant token embeddings, we conduct a simple empirical analysis in Figure 1. We tokenize each downstream dataset with the Qwen2.5 tokenizer and retrieve the entries from the input embedding matrix. As shown in Figure 1a, the corpus-level Ollivier–Ricci curvature distributions are dominated by negative values while retaining non-negligible positive-curvature mass, indicating heterogeneous local geometry. Figure 1b further shows that token embedding norm distributions exhibit non-trivial, domain-dependent patterns beyond a simple log-linear form, consistent with recent observations on structured token-embedding geometry (He et al., 2025a). These observations suggest that simply replacing Euclidean space with a single non-Euclidean geometry is insufficient; instead, token representations require a framework that can accommodate heterogeneous local geometric structures.

Limitations of Existing Methods. Nevertheless, efficiently integrating multiple geometries into LLM fine-tuning remains challenging. First, a single geometry may provide a useful but incomplete inductive bias when task-relevant tokens exhibit diverse local structures. Second, many non-Euclidean neural architectures rely on repeated exponential and logarithmic maps between Euclidean and curved spaces (Ganea et al., 2018; Yang et al., 2022), which can introduce additional computational overhead especially for LLMs. Therefore, an effective method should not only support heterogeneous geometric modeling, but preserve the efficiency expected from parameter-efficient adaptation.

To address these fundamental limitations, we propose a unified Mixture of Space (MoS) scheme that integrates three types of constant-curvature spaces (hyperbolic, spherical, and Euclidean), capturing diverse geometric structures within a single model simultaneously.

Building upon this framework, we introduce MoSLoRA, which combines the MoS paradigm with Low-Rank Adaptation (LoRA) for LLMs Parameter-Efficient Fine-Tuning (PEFT) of LLMs. MoSLoRA employs a lightweight routing mechanism that dynamically assigns each token to its optimal geometric expert through a unified projection with a novel grouped auxiliary loss, avoiding the computational overhead of repeated space transformations (commonly adopted in non-Euclidean methods) while maintaining the representational benefits of multiple geometries. This design enables the model to adapt its geometric inductive biases on-the-fly, matching the heterogeneous structural requirements of real-world language data. Our contributions can be summarized as follows:

- We propose MoSLoRA, a unified PEFT framework that couples LoRA with a mixture of geometric experts instantiated with Euclidean, Hyperbolic, and Spherical inductive biases.
- We design a lightweight routing with unified projection for efficient non-Euclidean space transformations, and introduce a grouped auxiliary loss for within-space expert balancing.
- We provide an in-depth analysis of the training dynamics of space selection and routing strategies, along with optimizing geometric space integration during fine-tuning.

- Experiments on natural language understanding and mathematical reasoning benchmarks show that MoSLoRA achieves higher average accuracy than strong baselines.

2 Related Work

Mixture of LoRA Experts. Mixture of Experts (MoE) introduces multiple expert networks and a gating network that selects experts based on different data characteristics (Jacobs et al., 1991; Shazeer et al., 2017), to improve the capacity and computational efficiency of large-scale models through combinations of experts. Recent work has begun to equip PEFT with MoE-style sparsity, yielding a line of LoRA-based mixture methods (Mangrulkar et al., 2022; Gao et al., 2022; Zadouri et al., 2024; Feng et al., 2024; Wu et al., 2024b;a; Dou et al., 2024; Zou et al., 2025). Representative ones include MELoRA (Ren et al., 2024), which constructs multiple LoRA experts by diagonal partitioning, HydraLoRA (Tian et al., 2024), which shares one LoRA factor (W_A) across experts while keeping multiple expert-specific factors (W_B), and HMoRA (Liao et al., 2025), which introduces hierarchical routing for multi-task settings. Despite their gains, these methods typically rely on homogeneous LoRA experts and often face a practical trade-off between performance and efficiency: stronger accuracy tends to require activating more experts (or higher effective rank), which increases compute and dilutes the sparsity benefits. In contrast, we extend LoRA experts to heterogeneous geometric spaces, and achieve a better accuracy-efficiency balance.

Non-Euclidean and Curvature-Aware Modeling. Recent research has increasingly explored non-Euclidean geometry to overcome the limitations of flat Euclidean representations (Shimizu et al., 2020; Peng et al., 2021; Pal et al., 2024; He et al., 2025b;a). Early studies developed fully hyperbolic neural networks by formulating neural operations directly in curved spaces (Chen et al., 2022), while mixed-curvature and product-space methods further combined multiple constant-curvature manifolds to model heterogeneous geometric structures (Skopek et al., 2020; Gu et al., 2018; Di Giovanni et al., 2022). Complementary to such static representation-space composition, we instantiate different constant-curvature spaces as routable experts, enabling adaptive geometric selection. More recently, curvature-aware modeling has also been extended to Transformer and LLM settings, including hyperbolic Transformers (Yang et al., 2024c), hyperbolic fine-tuning (Yang et al., 2024b), and hyperbolic LLMs with mixture-of-curvature experts (He et al., 2026). However, some of these LLM-oriented methods still rely on a single geometry (e.g., hyperbolic), while others focus on pretraining and are thus orthogonal to our work; we fill this gap by integrating hyperbolic, spherical, and Euclidean spaces simultaneously in a parameter-efficient expert framework.

3 Preliminary

Mixture of LoRA Experts. Mixture of LoRA Experts consists of a group of N structurally homogeneous experts $\{E_i\}_{i=1}^{N}$, each equipped with an independent LoRA module. Given a token embedding $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$, the output of the i-th LoRA expert is $\mathbf{e}_i = B_i A_i \mathbf{x}$, where $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$ are trainable low-rank matrices, and $r \ll \min\{d_{\text{in}}, d_{\text{out}}\}$ is the LoRA rank, with matrix A_i randomly initialized and matrix B_i set to all zero. The output of a mixture of LoRA experts is formulated as:

$$\mathbf{o} = W\mathbf{x} + \sum_{i=1}^{N} r_i(\mathbf{x})\mathbf{e}_i = W\mathbf{x} + \sum_{i=1}^{N} r_i(\mathbf{x}) B_i A_i \mathbf{x}, \tag{1}$$

where $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the frozen pretrained weight of the Feed-Forward Neural Network (FFN) block, $r(\mathbf{x})$ is the token router in the MoE module which routes each token into several distinct experts (e.g., top-K experts for sparse MoE (Fedus et al., 2022)) among all N experts.

Lorentz Model of Hyperbolic Space. The Lorentz model, also called the hyperboloid model, provides one of the isometric realizations of the hyperbolic space as a Riemannian manifold. Formally, an n-dimensional Lorentz model with constant negative curvature $\kappa < 0$ is defined as:

$$\mathbb{L}^{n,\kappa} = \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \;\middle|\; \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = 1/\kappa,\ x_t > 0 \right\},$$

where $\mathbf{x} = [x_t;\, \mathbf{x}_s]^\top$ with $x_t \in \mathbb{R}$ and $\mathbf{x}_s \in \mathbb{R}^n$, and the Lorentzian inner product is given by:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_t y_t + \mathbf{x}_s^\top \mathbf{y}_s = \mathbf{x}^\top \operatorname{diag}(-1, 1, \ldots, 1)\, \mathbf{y}.$$

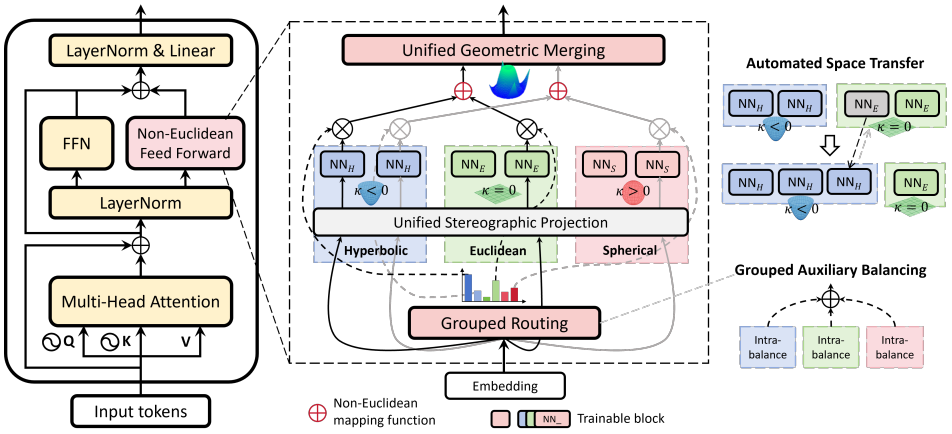


Figure 2: The MoSLoRA architecture contains heterogeneous geometric experts with various curvatures unified in our MoS scheme. Three geometric expert groups are embedded into the FFN layer and labeled in different colored blocks. The grouped auxiliary balancing enforces balanced routing within each space group while allowing free inter-space transitions, and remains fully reversible.

Geometrically, $\mathbb{L}^{n,\kappa}$ corresponds to the upper sheet of a two-sheeted hyperboloid embedded in $(n+1)$-dimensional Minkowski space, with the distinguished coordinate x_t representing the time-like axis and the remaining n coordinates forming the space-like axes. This construction not only aligns with the terminology of special relativity (Resnick, 1991) but also ensures numerical stability.

4 Mixture of Space Experts Framework

In this section, we present the Mixture of Space Experts framework, which adapts PEFT models to the implicit curvature of semantic subspaces and enables flexible exploration of different constant-curvature geometries within and across different subspaces.

4.1 Mixture of Space Scheme

We explore three constant-curvature spaces with positive, negative, and zero curvature, commonly modeled as spherical $\mathbb{S}$, hyperbolic $\mathbb{H}$, and Euclidean $\mathbb{E}$ spaces. For a n-dimensional hyperbolic space $\mathbb{H}^n_\kappa$ with constant negative curvature κ, each point $\mathbf{x} \in \mathbb{R}^{n+1}$ in $\mathbb{H}^n_\kappa$ should satisfy:

$$\mathbb{H}^n_\kappa := \{\mathbf{x} \in \mathbb{R}^{n+1} | \langle \mathbf{x}, \mathbf{x} \rangle_L = 1/\kappa, \kappa < 0\}, \tag{2}$$

where $\mathbb{H}^n_\kappa$ is defined by the Lorentz inner product $\langle \mathbf{x}, \mathbf{x} \rangle_L = -x_1^2 + \sum_{i=2}^{n+1} x_i^2$, and $\mathbf{x} = [x_1, x_2, \ldots, x_{n+1}] = [x_{\text{time}}, \mathbf{x}_{\text{space}}]$ denotes an arbitrary point with time-like component x_1 and space-like component $[x_2, \ldots, x_{n+1}]$. In the Lorentz model of hyperbolic space, the volume V_κ grows exponentially with its radius r as $V_\kappa(r) \asymp \exp\left((n-1)\sqrt{-\kappa}\, r\right)$. Thus, the larger the magnitude of κ, the greater the curvature and the faster the volume expansion, allowing the space to accommodate more hierarchical structures with higher representational capacity. For spherical spaces with curvature $\kappa > 0$, each point should satisfy:

$$\mathbb{S}^n_\kappa := \{\mathbf{x} \in \mathbb{R}^{n+1} | \langle \mathbf{x}, \mathbf{x} \rangle_2 = 1/\kappa, \kappa > 0\}, \tag{3}$$

where $\langle \cdot, \cdot \rangle_2$ is the Euclidean inner product and $\langle \mathbf{x}, \mathbf{x} \rangle_2 = \sum_{i=1}^{n+1} x_i^2$. Unlike hyperbolic space, which is suitable for representing hierarchical structures, volume in spherical spaces grows more slowly than Euclidean space with respect to radius; they are especially better for modeling cyclic, periodic, or bounded structures, where global capacity is limited but dense local clustering and angular relationships are crucial. For the zero-curvature case, we recover the Euclidean space as:

$$\mathbb{E}^n := \{\mathbf{x} \in \mathbb{R}^n \mid \kappa = 0\}, \tag{4}$$

where $\mathbf{x}$ is not subject to any curvature-induced norm constraint. Then we will augment it into the ambient space $\mathbb{R}^{n+1}$ to align the dimensionality with the other two spaces.

To unify the three aforementioned spaces and associated constraints, we introduce a unified scheme, MoS, which preserves the local degrees of freedom and the flexibility of transformations across all three spaces. This formulation follows naturally from the Lorentz inner-product constraint of hyperbolic space and the Euclidean inner-product constraint of spherical space: given the space-like component $\mathbf{s}$, the auxiliary coordinate is determined by satisfying the corresponding curvature constraint. The formal definition is given as follows:

$$\mathrm{G}(\mathbf{x}) = \begin{bmatrix} \sqrt{\|\mathbf{s}\|^2 \cdot \mathbf{sgn}(-\kappa) + \varphi(\kappa)} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} \xi \\ \mathbf{s} \end{bmatrix} \tag{5}$$

$$\mathbf{sgn}(\kappa) := \begin{cases} -1, & \kappa < 0, \\ 1, & \kappa \geq 0, \end{cases} \quad \varphi(\kappa) := \begin{cases} 1/|\kappa|, & \kappa \neq 0, \\ 0, & \kappa = 0. \end{cases}$$

where point $\mathbf{x} = (\xi; \mathbf{s}^T)^T \in \mathbb{R}^{n+1}$, $\xi \in \mathbb{R}, \mathbf{s} \in \mathbb{R}^n$, and $\mathrm{G}(\mathbf{x})$ denotes the unified space transformation from different geometries with different signs of the curvature κ. We restrict transformation to the space-like component, as Lorentz geometry in $\mathbb{R}^{n+1}$ provides only n degrees of freedom, yielding better numerical stability. For $\kappa = 0$, this formulation corresponds to the Euclidean space: the additional coordinate ξ only serves to align $n+1$ dimensions across spaces, while the effective Euclidean representation remains the space-like component $\mathbf{s} = \mathbf{x}$. This scheme enables embeddings to be jointly learned across manifolds, capturing complementary geometric information in the context.

4.2 Unified Stereographic Projection

While existing mapping schemes are capable of strictly preserving hierarchical structures (Chen et al., 2021; Bdeir et al., 2023; Yang et al., 2024b), they depend on computationally expensive and GPU-unfriendly operations, such as exponential and logarithmic maps (detailed in Appendix A), and therefore lack a unified, end-to-end formulation. To address this challenge, we propose a unified stereographic projection, as illustrated in the middle of Figure 2, which projects inputs from the input space into three constant-curvature spaces and subsequently maps them back to the output space. In our unified stereographic projection, each point $\mathbf{x}$ (from input space) is firstly transmitted into a projected space through stereographic conformal inverse projection $\rho_\kappa^{-1}(\cdot)$ in Eq. 6 to $(\xi; \mathbf{s}^T)^T \in \mathbb{R}^{n+1}$, where $\kappa \in \mathbb{R}$ is the curvature of the projected embedding space:

$$\rho_\kappa^{-1}(\mathbf{x}) = \left(\frac{1}{\sqrt{|\kappa|}} \frac{1 - \kappa\|\mathbf{x}\|_2^2}{1 + \kappa\|\mathbf{x}\|_2^2}, \frac{2\mathbf{x}}{1 + \kappa\|\mathbf{x}\|_2^2} \right)^T = \begin{bmatrix} \xi \\ \mathbf{s} \end{bmatrix} \tag{6}$$

Then, after model-related transformation from $(\xi; \mathbf{s}^T)^T$ into $(\xi'; \mathbf{s'}^T)^T$, tokens will be projected back to the original space by stereographic projection:

$$\rho_\kappa((\xi'; \mathbf{s'}^T)^T) = \mathbf{s}'/(1 + \sqrt{|\kappa|}\xi'), \tag{7}$$

To guarantee constraint satisfaction and numerical stability, we apply a scaling factor before the inverse projection and rescale the output afterward. This scaling controls the norm of projected embeddings and prevents them from approaching unstable boundary regions. In Proposition 4.1, we show that such scaling does not change the mapping, but is equivalent to a curvature reparameterization.

Proposition 4.1 (Scale-Invariant Equivalence of the Unified Projection)**.** *Let $\kappa \neq 0$, $\gamma > 0$, and $F_\kappa(\cdot)$ denote the unified projection and MoS defined in Eq. 5, 6, and 7. Applying a scaling factor γ to the input before the inverse projection and rescaling the output by the same factor yields:*

$$F_\kappa(x) \;=\; 1/\gamma \cdot F_{\kappa/\gamma^2}(\gamma x).$$

This equivalence justifies scaling as a stable implementation of the unified projection without altering its geometric effect. Since the projection avoids exponential and logarithmic maps and only uses norm-based and element-wise operations, it is more GPU-friendly for large Transformer models, as empirically verified in Figure 4. A detailed proof is provided in Appendix E.

4.3 MIXTURE OF SPACE PARAMETER-EFFICIENT TUNING

Building upon the unified geometric formulation introduced earlier in Eq. 5, we now specialize it to a mixture-of-experts PEFT setting and introduce our MoSLoRA. Specifically, given a projected embedding $(\xi_i; \mathbf{s}_i^T)^T \in \mathbb{R}^{n+1}$ of token $\mathbf{x}_i$, we treat the space-like component $\mathbf{s}_i \in \mathbb{R}^n$ as the routing unit for MoSLoRA within each expert, as illustrated in Figure 2. Each space expert is defined as:

$$\mathrm{G}(W, \mathbf{x}_i) = \begin{bmatrix} \sqrt{\|W\mathbf{s}_i\|^2 \cdot \mathbf{sgn}(-\kappa) + \varphi(\kappa)} \\ W\mathbf{s}_i \end{bmatrix} = \begin{bmatrix} \xi_i' \\ \mathbf{s}_i' \end{bmatrix} \tag{8}$$

The output token embeddings $(\xi'_{t,e}; {\mathbf{s}'_{t,e}}^T)^T \in \mathbb{R}^{n+1}$ produced by the top-K geometric experts are projected back into the Euclidean space, where the relations captured in different curved spaces are preserved and aggregated. The non-Euclidean feed-forward is formulated as follows:

$$\mathbf{o}_j = \sum_{i=1}^{N} r_i\left(\mathbf{x}_j\right) \rho_{\kappa_i}((\xi_j; {\mathbf{s}'_j}^T)^T), \tag{9}$$

where $\rho_{\kappa_i}((\cdot;\cdot))$ is the stereographic projection to map points (token embeddings) from the curved space back to the output space, κ_i is the curvature of the i-th geometric expert, $r_i(x_j)$ is the routing value for token x_j and i-th expert. It is worth noting that the curvature κ of each expert is a learnable parameter, allowing the model to independently adjust the sub-space of each expert to approach the optimal geometry. From an efficiency perspective, the stereographic mapping and the associated routing strategy used here require neither complex exponential/logarithmic space operations nor prior knowledge of the latent embedding space, thereby yielding higher efficiency in both training and inference. We further establish the following Proposition 4.2, with proof provided in Appendix D. This result shows that the proposed geometric expert remains well-conditioned under the valid curvature-domain constraints and does not introduce unbounded gradients during training.

Proposition 4.2 (Bounded Gradient of MoSLoRA Experts)**.** *Let $z = Ws_i$ denote the transformed space-like component in Eq. 8, and let ξ_i' denote the first coordinate of the geometric expert output in Eq. 8. Then, $\|\nabla_z \xi_i'\|_2$ is bounded by 1 for $\kappa < 0$ and by 0 for $\kappa = 0$. For $\kappa > 0$, if the spherical-domain constraint is satisfied with a margin, i.e., there exists a constant R such that $\|z\|_2 \le R < 1/\sqrt{\kappa}$, then $\|\nabla_z \xi_i'\|_2 \le R/\sqrt{1/\kappa - R^2}$.*

Grouped Auxiliary Loss. To stabilize routing while preserving the specialization of different geometric spaces, we introduce a novel grouped auxiliary loss (in Figure 2: grouped routing) to encourage balancing within the same space. Let $\mathcal{G}$ denote the set of geometric space groups, and let $\mathcal{E}_g \subseteq \{1, \ldots, N\}$ denote the experts assigned to group $g \in \mathcal{G}$. Following the standard MoE, P_i and F_i denote the average routing probability and the selected-token fraction of expert i, respectively. Instead of balancing all experts globally, we perform load balancing within each geometric group:

$$\mathcal{L}_{\text{aux}}^{\text{group}} = \alpha \sum_{g \in \mathcal{G}} |\mathcal{E}_g| \sum_{i \in \mathcal{E}_g} P_i F_i, \tag{10}$$

where α is the auxiliary loss coefficient. This objective prevents expert collapse within each space while avoiding unnecessary competition across different spaces.

Since curvature parameters control the geometry of each expert and differ from capacity-related LoRA weights, we optimize them with a separated optimizer. This design enables more flexible curvature exploration without being dominated by the optimization dynamics of other trainable parameters. Detailed parameter grouping and update rules are provided in Appendix C.3.

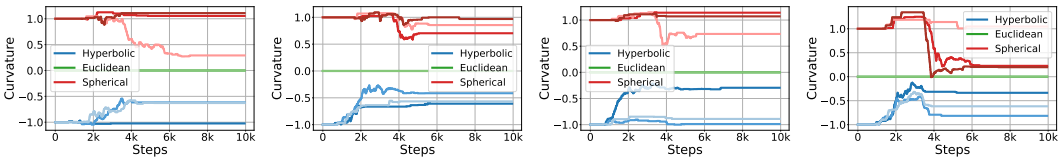
5 EXPERIMENTS

5.1 EXPERIMENT SETTING

Dataset and Benchmarks. To explore the utility of our method, we evaluate our approach on both natural language understanding (Wang et al., 2018) and mathematical reasoning datasets. For NLP tasks, we adopt the Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005), Recognizing Textual Entailment (RTE) (Bowman et al., 2015) and the Corpus of Linguistic

Table 1: Performance comparison of different tuning methods across different natural language understanding and mathematical reasoning benchmarks based on Qwen2-1.5B. Avg. reports the average accuracy, and rank ↑ indicates that a larger rank parameter is used compared to other methods.

Methods	COLA	MRPC	GSM8k	MATH500	MAWPS	SVAMP	AQuA	Avg.
LoRA	87.25	87.30	44.58	15.60	63.85	63.33	28.35	55.75
DoRA	87.15	87.19	42.53	14.20	62.12	62.33	30.71	55.18
LoRA rank ↑	83.51	86.43	46.85	13.60	61.92	60.33	**33.46**	55.16
AdaLoRA	82.45	79.77	**51.33**	16.60	62.50	58.67	23.23	53.51
MELoRA	86.96	86.55	44.96	16.40	62.69	**65.67**	32.28	56.50
HMoRA	66.63	66.49	12.28	2.60	54.23	46.00	24.41	38.95
HypLoRA	85.91	87.71	45.56	14.60	62.88	64.67	27.95	55.61
HydraLoRA	87.54	88.17	43.67	16.80	62.88	62.67	28.35	55.73
MoSLoRA (Ours)	**87.63**	**88.23**	47.76	**18.00**	**74.62**	64.33	31.50	**58.87**



(a) Layer-1 (b) Layer-10 (c) Layer-15 (d) Layer-20

Figure 3: Curvature dynamics of each geometric expert of different layers during training.

Acceptability (CoLA) (Warstadt et al., 2019). For mathematical reasoning, we employ GSM8K (Cobbe et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), and AQuA (Ling et al., 2017), alongside the MATH500 by OpenAI (Lightman et al., 2023), a curated challenging subset of the MATH benchmark (Hendrycks et al., 2021). For commonsense reasoning, we include OpenBookQA (OBQA) (Mihaylov et al., 2018) and CommonsenseQA (CSQA) (Talmor et al., 2019). The training set is constructed by uniformly sampling from the mathematical datasets except MATH500, which is reserved for evaluating zero-shot performance on unseen problems.

Base Models and Baselines. We include Qwen2-1.5B (Yang et al., 2024a), Qwen2.5-3B, Qwen2.5-7B (Qwen et al., 2025), and Gemma2-2B-it (Team et al., 2024) as our base models. For PEFT methods, we compare MoSLoRA with LoRA (Hu et al., 2022), AdaLoRA (Zhang et al., 2023), DoRA (Liu et al., 2024), MELoRA (Ren et al., 2024), HMoRA (Liao et al., 2025), HydraLoRA (Tian et al., 2024) and HypLoRA (Yang et al., 2024b). These methods share the same training and evaluation settings as ours, and some incorporate architectures related to the MoE design. Here we adopt an asymmetric architecture similar to Tian et al. (2024), minimizing the activated parameters while verifying the effectiveness of our proposed MoSLoRA framework.

Training Settings. All models are trained for three epochs on the same dataset using NVIDIA A100 and H800 GPUs. To ensure fairness, we control the number of activated parameters across methods. More detailed information can be found in Appendix B.1.

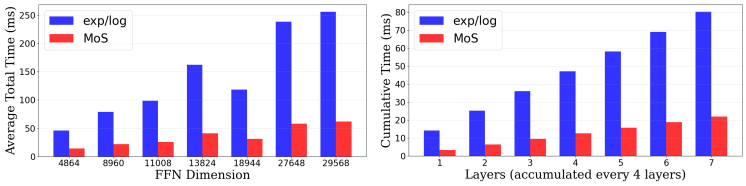
5.2 Baselines Comparison

We compare our methods with other baselines using 8 distinct experts with top-4 routing. In MoSLoRA, three experts are allocated for each non-Euclidean space group and two for the Euclidean group, with curvatures initialized to -1 for negative curvature and 1 for positive curvature spaces. To keep the number of activated parameters comparable across methods, we vary the LoRA rank from 8 to 64 (0.45%–3.53% of activated parameters), while fixing the rank at 8 for all other baselines. Due to their fully activated computation, HMoRA and HydraLoRA yield 2.26% and 2.54% activated parameters, respectively, whereas MoSLoRA requires only 1.31%.

Main Results. As shown in Table 1, our MoSLoRA achieves higher average performance than existing state-of-the-art methods in both natural language understanding and mathematical reasoning benchmarks. In particular, it obtains the best results on most mathematical reasoning benchmarks,

Table 2: Performance comparison on math and commonsense reasoning, and language understanding benchmarks based on Qwen2.5-3B.

Methods	GSM8k	MATH500	MAWPS	SVAMP	AQuA	MRPC	CoLA	RTE	CSQA	OBQA	Avg.
Base	59.06	0.40	60.19	74.67	50.00	56.41	62.80	51.62	32.40	32.40	47.99
LoRA	**65.50**	25.60	65.58	76.33	38.98	87.71	69.13	89.53	82.80	87.00	68.81
HydraLoRA	64.52	22.40	65.77	75.33	**42.91**	**88.46**	85.91	89.89	**82.56**	**89.20**	70.69
HypLoRA	64.67	26.40	65.77	78.67	41.34	88.23	**87.34**	90.25	79.85	87.80	71.03
MoSLoRA	64.06	**30.80**	**81.15**	**79.67**	40.94	88.41	86.19	**90.97**	82.47	89.00	**73.37**



(a) Increasing layer dimensions. (b) Increasing model layers.

Figure 4: Efficiency comparison between exponential and logarithmic operations and our MoS.

including the challenging MATH500. Since our training data is uniformly sampled from other mathematical reasoning datasets, these results further demonstrate the superior generalization ability of our MoSLoRA when tackling previously unseen reasoning problems. We observe that our method achieves particularly notable improvements on mathematical reasoning benchmarks compared to existing baselines. We attribute this to the introduction of the mixed-curvature framework, where hyperbolic space is especially well suited for representing hierarchical numerical and logical structures (Yang et al., 2024b), while spherical space can better capture cyclic properties such as equivalence relations commonly present in mathematics. Consequently, our approach provides clear advantages in handling large-number computations and symbolic operations. We also scale to 3B models (other architectures and 7B results are in Appendix C.1 and C.2) and more benchmarks to further validate the effectiveness. As shown in Table 2, MoSLoRA demonstrates strong competitiveness across all evaluated benchmarks, and achieving the best overall average performance among all baselines.

5.3 Tuning Dynamics

Curvature Dynamics. The curvature parameter κ in MoSLoRA is different from other capacity parameters, serving as a key geometric property that simultaneously characterizes both the input space and the model's embedding space. As shown in Figure 3, we track the evolution of embedding spaces for geometric experts across model layers. Distinct colors denote different space types, with varying intensities representing spaces with different curvatures. To maintain stable Euclidean embeddings, two Euclidean experts with zero curvature were included in all experiments. During training, the geometric experts progressively selected the curvature spaces that minimized the loss, and after approximately 6k–8k steps these selections stabilized, suggesting that the model had automatically identified the optimal mixture of embedding spaces. We also observed interesting dynamic behaviors, such as recurrent switching across curvature spaces and transitions between geometric spaces. For instance, in Figure 3d, one spherical expert in the 20th layer temporarily shifted into the hyperbolic space around step 4k, but subsequently returned to the spherical space and ultimately converged to a positive curvature near 0.2. These dynamics indicate that MoSLoRA can adaptively adjust combinations of geometric experts across spaces, highlighting the flexibility and stability of our MoS architecture. Notably, the observed cross-space transitions suggest that the grouped auxiliary loss does not force a fixed global balance across all experts; instead, it stabilizes expert utilization within each geometric group while preserving flexible selection among different spaces. Furthermore, layer-wise trends reveal that lower layers, particularly the first layer, exhibit more stable but slower convergence compared to higher layers. This aligns with observations in MoE-related literature (Dai et al., 2024; Muennighoff et al., 2024): early layers primarily capture task-agnostic token-level information, leading to smoother geometry selection, while higher layers encode task-specific information with more complex curvature patterns.

Table 3: Performance comparison of single-space variants across math and reasoning benchmarks.

Methods	Space	COLA	MRPC	GSM8k	MATH500	MAWPS	SVAMP	AQuA	Avg.
MoSLoRA-S	Spherical	86.86	87.77	44.50	17.60	72.69	**64.33**	28.35	57.44
MoSLoRA-H	Hyperbolic	85.81	**89.33**	45.34	17.80	73.46	63.67	**34.25**	58.52
MoSLoRA-E	Euclidean	87.15	87.19	42.23	16.20	70.77	61.67	31.89	56.73
MoSLoRA	Mixture	**87.63**	88.23	**47.76**	**18.00**	**74.62**	**64.33**	31.50	**58.87**

Training Efficiency. In Figure 4, we compare the runtime between our proposed lightweight space mapping method and the exp/log mappings. Figure 4a illustrates how computation time scales with the increasing layer dimension, while Figure 4b shows the trend related to the number of model layers. The results show that MoS achieves significant speedups up to 4× over exp/log, and this relative acceleration remains consistent as the network depth increases and the embedding dimension grows. Detailed comparisons of memory overhead and average training step time are provided in Appendix F, showing that MoSLoRA improves computational efficiency compared to common non-Euclidean approaches, offering insights for exploring efficient solutions for non-Euclidean models.

5.4 Ablation Study

Single-Space Variants. To examine whether the performance gain comes from heterogeneous geometric spaces rather than the MoE structure, we compare the full MoSLoRA with its single-space variants. All variants follow the same MoE routing framework, top-4 out of 8 expert selection, and auxiliary balancing loss, while differing only in the geometric type of experts. As shown in Table 3, the mixed-space MoSLoRA achieves the best average performance, outperforming the all single-space variants. Meanwhile, different variants show advantages on different datasets, e.g., Hyperbolic experts perform strongly on MATH500 and AQuA, Euclidean experts are competitive on CoLA, and Spherical experts are effective on SVAMP. These results indicate that no single geometry is uniformly optimal, and the benefit comes from combining all geometric spaces.

Table 4: Different geometric expert allocation strategies.

Methods	GSM8k	MATH500	MAWPS	SVAMP	AQuA	Avg.
H4-E2-S2	43.21	17.60	**75.20**	**66.33**	**34.25**	**47.32**
H2-E2-S4	42.38	17.60	73.84	64.33	29.53	45.54
H3-E2-S3	**47.76**	**18.00**	74.62	64.33	31.50	47.24
H2-E4-S2	44.28	16.60	73.30	62.00	27.95	44.83

Different Space Experts Allocation. We further analyze how the number of experts assigned to each geometric space affects performance. In Table 4, H, E, and S denote Hyperbolic, Euclidean, and Spherical experts, respectively. For instance, H3-E2-S3 contains 3 Hyperbolic, 2 Euclidean, and 3 Spherical experts, which is the default setting in previous experiments. The results show that increasing the proportion of hyperbolic experts leads to a clear performance gain: H4-E2-S2 improves over H3-E2-S3 on most benchmarks. This observation is consistent with curvature analysis in Figure 1a, where token embeddings exhibit dominant negative curvature while still retaining non-negligible positive-curvature mass. Therefore, a larger hyperbolic expert proportion can better match the overall geometry of the data, while the spherical experts are still essential. These results further demonstrate the flexibility and potential of MoSLoRA under different geometric expert allocations.

6 Conclusion

In this work, we introduce MoS, a unified framework that integrates distinct geometric spaces and enables flexible transformations among three constant-curvature spaces: Hyperbolic, Euclidean, and Spherical spaces. Building on this framework, we further propose MoSLoRA, a PEFT method that combines experts within different geometric spaces. This design allows LLMs to dynamically adjust the curvature of each expert's underlying space during fine-tuning and to flexibly reconfigure combinations of different spaces based on the input. In addition, to address the computational overhead of exponential and logarithmic operations commonly adopted in existing non-Euclidean models, we develop a lightweight routing and space-mapping strategy that improves the efficiency of space transitions. Experiments demonstrate that MoSLoRA outperforms strong baselines and provides new insights into geometric representation learning. Future work may extend MoSLoRA to industrial-scale applications and reinforcement learning settings.

AI USE STATEMENT

In this work, we used generative AI tools only for recommended-disclosure tasks: editing the paper to improve readability, formatting references, and polishing LaTeX. We have not used generative AI tools for any required-disclosure task, including generating synthetic data; developing theoretical models or conceptual frameworks; formulating mathematical claims or proof ingredients; assisting with proofs; proposing or refining hypotheses; designing or giving feedback on methodology/experiments; implementing methods; translation; cleaning or reformatting datasets; qualitative/thematic data analysis; or interpreting results; the remaining required-disclosure tasks are not applicable to this work. We reviewed all AI-assisted edits and verified that the final text, claims, code, data, and results are accurate and our own. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code and datasets used in this work are available at the anonymous repository `https://anonymous.4open.science/r/MoSLoRA`. The detailed experimental configurations and evaluation protocols are described in the experiments section of the main text, and further training details are provided in Appendix B. All results in this paper can be reproduced using the released code and the provided configurations.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. *arXiv preprint arXiv:2303.15919*, 2023.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1075. URL `https://aclanthology.org/D15-1075/`.

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

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

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. *arXiv preprint arXiv:2105.14686*, 2022. doi: 10.48550/arXiv.2105.14686. ACL 2022 Main Conference.

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

Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.70. URL `https://aclanthology.org/2024.acl-long.70/`.

Francesco Di Giovanni, Giulia Luise, and Michael Bronstein. Heterogeneous manifolds for curvature-aware graph embedding. *arXiv preprint arXiv:2202.01185*, 2022.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*, 2005.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1932–1945, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-loras: An efficient multitask tuning for large language models. *arXiv preprint arXiv:2403.03432*, 2024.

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

Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen. Parameter-efficient mixture-of-experts architecture for pre-trained language models. *arXiv preprint arXiv:2203.01104*, 2022.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International conference on learning representations*, 2018.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Neil He, Jiahong Liu, Buze Zhang, Ngoc Bui, Ali Maatouk, Menglin Yang, Irwin King, Melanie Weber, and Rex Ying. Position: Beyond euclidean–foundation models should embrace non-euclidean geometries. *arXiv preprint arXiv:2504.08896*, 2025a.

Neil He, Menglin Yang, and Rex Ying. Lorentzian residual neural networks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, pp. 436–447. Association for Computing Machinery, 2025b. ISBN 9798400712456.

Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassiulas, Menglin Yang, and Rex Ying. Helm: Hyperbolic large language models via mixture-of-curvature experts. *Advances in Neural Information Processing Systems*, 38:142604–142635, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 1991.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 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.

Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. Hmora: Making llms more effective with hierarchical mixture of lora experts. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation : Learning to solve and explain algebraic word problems, 2017. URL https://arxiv.org/abs/1705.04146.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. *arXiv preprint arXiv:2410.01131*, 2024.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

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

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

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

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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

Robert Resnick. *Introduction to special relativity*. John Wiley & Sons, 1991.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

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

Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*, 2020. doi: 10.48550/arXiv.1911.08411.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37: 9565–9584, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024a.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024b.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. 2024a. URL https://arxiv.org/abs/2407.10671.

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

Menglin Yang, Aosong Feng, Bo Xiong, Jihong Liu, Irwin King, and Rex Ying. Hyperbolic fine-tuning for large language models. *arXiv preprint arXiv:2410.04010*, 2024b.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient hyperbolic transformer fully in hyperbolic space. In *Proceedings of the 2024 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2024c.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. Flylora: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. *arXiv preprint arXiv:2510.08396*, 2025.

A TANGENT SPACE, EXPONENTIAL AND LOGARITHMIC MAPS.

For each point $\mathbf{x} \in \mathbb{L}^{n,\kappa}$, tangent space $\mathcal{T}_{\mathbf{x}}\mathbb{L}^{n,\kappa}$ is defined as the Lorentz-orthogonal complement of $\mathbf{x}$ and constitutes a smooth Euclidean subspace of $\mathbb{R}^{n+1}$. Tangent space provides a local linear approximation of the curved manifold for optimization and representation learning. The transition between the manifold and its tangent space is realized through exponential and logarithmic maps. The exponential map $\exp_{\mathbf{x}}^{\kappa} : \mathcal{T}_{\mathbf{x}}\mathbb{L}^{n,\kappa} \to \mathbb{L}^{n,\kappa}$ takes a tangent vector $\mathbf{u} \in \mathcal{T}_{\mathbf{x}}\mathbb{L}^{n,\kappa}$ and projects it onto the manifold along the geodesic starting at $\mathbf{x}$, which is:

$$\exp_{\mathbf{x}}^{\kappa}(\mathbf{u}) = \cosh\Big(\sqrt{|\kappa|}\|\mathbf{u}\|_{\mathcal{L}}\Big)\,\mathbf{x} + \frac{\sinh\Big(\sqrt{|\kappa|}\|\mathbf{u}\|_{\mathcal{L}}\Big)}{\sqrt{|\kappa|}\|\mathbf{u}\|_{\mathcal{L}}}\,\mathbf{u}. \tag{11}$$

Conversely, the logarithmic map $\log_{\mathbf{x}}^{\kappa} : \mathbb{L}^{n,\kappa} \to \mathcal{T}_{\mathbf{x}}\mathbb{L}^{n,\kappa}$ takes a point $\mathbf{y} \in \mathbb{L}^{n,\kappa}$ and returns the unique tangent vector at $\mathbf{x}$ that corresponds to the geodesic connecting $\mathbf{x}$ and $\mathbf{y}$:

$$\log_{\mathbf{x}}^{\kappa}(\mathbf{y}) = \frac{\cosh^{-1}\big(\kappa\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\big)}{\sinh\big(\cosh^{-1}(\kappa\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}})\big)}\left(\mathbf{y} - \kappa\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\mathbf{x}\right). \tag{12}$$

These two maps establish a rigorous correspondence between the locally Euclidean tangent space and the globally curved hyperbolic manifold.

B TRAINING DETAILS

B.1 TRAINING HYPERPARAMETERS

Table 5 presents the hyperparameters used to fine-tune the models with MoSLoRA on language understanding, commonsense reasoning and mathematical reasoning. The same hyperparameter settings are applied across all tasks. Each experiment is conducted independently, with a single run for each model. The final model trained is used for evaluation. For the baseline methods, the same hyperparameter configuration is reused.

Table 5: Hyperparameters for MoSLoRA.

Hyperparameter	**Value**
Num Train Epoch	3
Optimizer	AdamW
Weight Decay	0.01
Warmup Ratio	0.1
Learning Rate	3×10^{-4}
Projection Scaling γ	0.001
Target Modules	`gate_proj`, `down_proj`, `up_proj`

B.2 STATISTICS OF THE LANGUAGE UNDERSTANDING DATASET

We conduct experiments using a subset of the General Language Understanding Evaluation (GLUE) dataset (Wang et al., 2018), a benchmark for training, evaluating, and analyzing natural language understanding systems. Specifically, we selected datasets Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005), Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019) and Recognizing Textual Entailment (RTE) (Bowman et al., 2015). As shown in Table 6, this dataset consists of various training and testing examples, each designed to evaluate specific linguistic tasks, including semantic equivalence and grammatical acceptability.

B.3 STATISTICS OF MATHEMATICAL REASONING DATASETS

As illustrated in the Table 7, we have constructed a mathematical reasoning training set consisting of a mix of four datasets, totaling 13,262 examples. These datasets include GSM8K (Cobbe et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), and a subset of AQuA (Ling

Table 6: The detailed statistics of language understanding datasets.

Dataset	Train	Test	Task Description
MRPC	3,668	1,725	Determine if a pair of sentences is semantically equivalent
CoLA	8,551	1,043	Evaluate the grammatical acceptability of English sentences
RTE	2,490	277	Recognize if a hypothesis is entailed by text

et al., 2017), each focusing on different aspects of mathematical reasoning. To control the contribution of AQuA, we do not use its full training split; instead, we uniformly sample 4,000 examples from AQuA's training set using a fixed random seed (42). Additionally, we have incorporated MATH500 (Lightman et al., 2023) into the test set to further evaluate the model's performance.

Table 7: The detailed statistics of mathematical reasoning datasets.

Dataset	Data Number	Task Type
Train	13,262	Mixed
Test		
GSM8K	1,319	Question-Answering
MAWPS	520	Question-Answering
SVAMP	300	Question-Answering
MATH500	500	Question-Answering
AQuA	254	Option

B.4 Statistics of commonsense reasoning datasets

Also, to extend the scope of our experiments beyond mathematical reasoning and language understanding, we further evaluate our approach on commonsense reasoning tasks. As shown in Table 8, we construct the commonsense reasoning training set using two widely adopted multiple-choice benchmarks: OpenBookQA (OBQA) (Mihaylov et al., 2018) and CommonsenseQA (CSQA) (Talmor et al., 2019). For training, we use the official training splits of both datasets. Evaluation is conducted on their respective validation and test sets to ensure consistency with prior work.

Table 8: The detailed statistics of commonsense reasoning datasets.

Dataset	Train	Test	Task Description
OBQA	4,957	500	Answer science questions using provided facts plus commonsense
CSQA	9,742	1,221	Choose correct option requiring diverse commonsense knowledge

C More Experiments

C.1 Main results on gemma2

To verify that the proposed method is not limited to a specific architectural design, we evaluate its performance across multiple parameter-efficient fine-tuning variants under the same backbone model Gemma2-2b-it, as shown in Table 9. This allows us to examine the robustness and generality of our approach across different architectural configurations.

C.2 Main results on larger models

To further evaluate the scalability of our approach, we extend the experiments beyond the 2B or 3B models and adopt Qwen2.5-7B as the base model. This setting allows us to examine whether MoSLoRA remains effective when applied to a larger backbone with substantially increased parameter capacity. As shown in Table 10, MoSLoRA still achieves the highest average accuracy 59.52% among all compared methods. Moreover, it obtains the best performance on most subtasks, demonstrating

Table 9: Comparison of parameter-efficient fine-tuning methods across reasoning and language understanding benchmarks. All results are reported on the same backbone model (Gemma2-2b-it).

Method	GSM8k	MATH500	MAWPS	SVAMP	AQuA	MRPC	CoLA	RTE	Avg.
LoRA	**52.24**	13.80	71.33	65.19	**32.68**	89.39	84.66	90.25	62.44
AdaLoRA	46.70	7.20	64.23	62.67	20.08	84.17	83.03	87.36	56.93
DoRA	48.60	11.20	71.67	64.62	31.89	**89.45**	85.33	89.89	61.58
MELoRA	48.90	11.80	63.85	70.00	28.35	88.75	84.08	88.81	60.57
HypLoRA	45.34	10.80	64.23	66.33	27.56	87.25	84.08	87.36	59.12
HydraLoRA	51.18	**14.80**	65.38	71.33	28.35	88.58	**86.29**	**90.97**	62.11
MoSLoRA	47.53	14.40	**81.35**	**73.00**	31.50	88.63	85.43	89.17	**63.88**

Table 10: Performance comparison of different tuning methods on larger scale model (Qwen2.5-7B).

Methods	GSM8k	MATH500	MAWPS	SVAMP	AQuA	Avg.
AdaLoRA	71.49	12.60	66.15	80.33	44.09	54.93
MELoRA	47.76	16.40	65.77	81.00	47.24	51.63
HMoRA	46.85	15.80	65.77	80.33	42.91	50.33
HydraLoRA	67.63	19.40	65.96	81.00	48.82	56.56
HypLoRA	**72.48**	18.00	66.15	83.33	42.52	56.50
MoSLoRA	52.46	**23.20**	**84.81**	**84.00**	**53.15**	**59.52**

Table 11: Performance of different optimizer choices across different MATH and general reasoning benchmarks. UNI and SEP denote using a unified optimizer and separated optimizers, respectively.

Methods	COLA	MRPC	GSM8k	MATH500	MAWPS	SVAMP	AQuA	Avg.
MoSLoRA UNI	87.15	87.19	46.85	17.80	74.62	64.33	29.13	58.15
MoSLoRA SEP	**87.63**	**88.23**	**47.76**	**18.00**	**74.62**	**64.33**	**31.50**	**58.87**

that the proposed multi-space geometric expert design remains effective and scalable under larger model sizes.

C.3 SEPARATED OPTIMIZER FOR CURVATURE PARAMETERS

We separate trainable parameters into curvature parameters and capacity-related parameters. Let

$$\Theta = \{\kappa^{(1)}, \ldots, \kappa^{(C)}, \theta^{(1)}, \ldots, \theta^{(M)}\}, \tag{13}$$

where $\kappa^{(c)}$ denotes the c-th curvature parameter group and $\theta^{(m)}$ denotes the m-th capacity-related parameter group. We use C rather than K to avoid confusion with top-K routing.

For a trainable parameter $w \in \Theta$, the separated update rule is

$$w \leftarrow w - \sum_{c=1}^{C} \eta_{\kappa}^{(c)} \mathbf{1}_{w \in \kappa^{(c)}} g_{\kappa}^{(c)}(w) - \sum_{m=1}^{M} \eta_{\theta}^{(m)} \mathbf{1}_{w \in \theta^{(m)}} g_{\theta}^{(m)}(w), \tag{14}$$

where $\eta_{\kappa}^{(c)}$ and $\eta_{\theta}^{(m)}$ are the learning rates for curvature and capacity-related parameter groups, respectively; $g_{\kappa}^{(c)}(\cdot)$ and $g_{\theta}^{(m)}(\cdot)$ denote the corresponding gradients; and $\mathbf{1}_{(\cdot)}$ is an indicator function selecting the parameter group.

We observed that when using a unified optimizer, the curvature κ fail to adjust alongside other parameters, resulting in nearly static values and relatively poor task performance (in Table 11). Therefore, we assign independent optimizers and learning rates to κ of all geometric experts, enabling the model to adjust latent geometric spaces more flexibly without being constrained by the optimization trajectory of other capacity-related parameters. Empirically, this setting leads to improved performance on downstream tasks.

D GRADIENT BOUND ANALYSIS

The following presents a gradient analysis of our MoSLoRA framework, demonstrating that the gradients in our design remain bounded.

Bound on the gradient w.r.t. u. Let $u = \phi(z)$ and define the lifting coordinate

$$a_\kappa(u) = \sqrt{\operatorname{sgn}(-\kappa)\,\|u\|^2 + \varphi(\kappa)}, \qquad \varphi(\kappa) = \begin{cases} 1/|\kappa|, & \kappa \neq 0, \\ 0, & \kappa = 0\,. \end{cases}$$

A direct differentiation gives

$$\nabla_u a_\kappa(u) = \frac{\operatorname{sgn}(-\kappa)\,u}{\sqrt{\operatorname{sgn}(-\kappa)\,\|u\|^2 + \varphi(\kappa)}}, \qquad \big\|\nabla_u a_\kappa(u)\big\| = \frac{\|u\|}{\sqrt{\operatorname{sgn}(-\kappa)\,\|u\|^2 + \varphi(\kappa)}}. \tag{15}$$

Case $\kappa < 0$ (hyperbolic). Here $\operatorname{sgn}(-\kappa) = 1$ and $\varphi(\kappa) = 1/|\kappa|$. Hence

$$\big\|\nabla_u a_\kappa(u)\big\| = \frac{\|u\|}{\sqrt{\|u\|^2 + 1/|\kappa|}} \leq 1 \quad \text{for all } u,$$

so a_κ is globally 1-Lipschitz in u.

Case $\kappa > 0$ (spherical). Now $\operatorname{sgn}(-\kappa) = -1$ and the domain requires $\|u\|^2 \leq 1/|\kappa|$. Let $R = \sup \|u\| < 1/\sqrt{|\kappa|}$, which can be enforced by a bounded activation together with scaling/projection. Then, from Eq. 15,

$$\big\|\nabla_u a_\kappa(u)\big\| = \frac{\|u\|}{\sqrt{1/|\kappa| - \|u\|^2}} \leq \frac{R}{\sqrt{1/|\kappa| - R^2}} =: C_u(\kappa, R) < \infty.$$

Therefore, as long as u stays at a fixed margin from the boundary (e.g., $R = (1-\varepsilon)/\sqrt{|\kappa|}$ with $\varepsilon > 0$), the gradient is uniformly bounded.

E SCALE EQUIVARIANCE OF THE UNIFIED GEOMETRIC EXPERT

We consider the full non-Euclidean feed-forward mapping defined by the composition

$$F_\kappa(x) := \rho_\kappa(G(W,\, x))\,,$$

where the mapping consists of the inverse stereographic projection, the unified geometric expert transformation, and the final projection back to Euclidean space. We show that this mapping is *scale-equivariant* under a coupled rescaling of the input and curvature.

Claim. For any scalar $\gamma > 0$ and nonzero curvature $\kappa \neq 0$,

$$F_\kappa(x) \;=\; \frac{1}{\gamma}\, F_{\kappa/\gamma^2}(\gamma x).$$

Proof. We denote the rescaled curvature by

$$\tilde{\kappa} := \frac{\kappa}{\gamma^2}.$$

Step 1: Scaling behavior of the space-like component (Eq. 5). The space-like component produced by the inverse stereographic projection is

$$s_\kappa(x) = \frac{2x}{1 + \kappa\|x\|^2}.$$

Evaluating this expression at the rescaled input γx with curvature $\tilde{\kappa}$, we obtain

$$\begin{aligned} s_{\tilde{\kappa}}(\gamma x) &= \frac{2(\gamma x)}{1+\tilde{\kappa}\|\gamma x\|^2} \\ &= \frac{2\gamma x}{1+(\kappa/\gamma^2)\,\gamma^2\|x\|^2} \\ &= \frac{2\gamma x}{1+\kappa\|x\|^2} \\ &= \gamma\, s_\kappa(x). \end{aligned}$$

Applying the linear transformation W yields

$$s'_{\tilde{\kappa}}(\gamma x) = W s_{\tilde{\kappa}}(\gamma x) = \gamma\, W s_\kappa(x) = \gamma\, s'_\kappa(x).$$

Step 2: Scaling behavior of the time-like component. The time-like coordinate is defined as

$$\xi' = \sqrt{\|s'\|^2\,\mathrm{sgn}(-\kappa) + \varphi(\kappa)}, \qquad \varphi(\kappa) = \frac{1}{|\kappa|}.$$

Note that

$$\mathrm{sgn}(-\tilde{\kappa}) = \mathrm{sgn}(-\kappa), \qquad \varphi(\tilde{\kappa}) = \frac{1}{|\tilde{\kappa}|} = \frac{\gamma^2}{|\kappa|} = \gamma^2\varphi(\kappa).$$

Therefore,

$$\begin{aligned} \xi'_{\tilde{\kappa}}(\gamma x) &= \sqrt{\|s'_{\tilde{\kappa}}(\gamma x)\|^2\,\mathrm{sgn}(-\kappa) + \varphi(\tilde{\kappa})} \\ &= \sqrt{\gamma^2\|s'_\kappa(x)\|^2\,\mathrm{sgn}(-\kappa) + \gamma^2\varphi(\kappa)} \\ &= \sqrt{\gamma^2\big(\|s'_\kappa(x)\|^2\,\mathrm{sgn}(-\kappa) + \varphi(\kappa)\big)} \\ &= \gamma\,\xi'_\kappa(x). \end{aligned}$$

Step 3: Scaling behavior of the projection. The final projection is given by

$$\rho_\kappa(\xi', s') = \frac{s'}{1+\sqrt{|\kappa|}\,\xi'}.$$

Substituting the scaled quantities yields

$$\begin{aligned} F_{\tilde{\kappa}}(\gamma x) &= \rho_{\tilde{\kappa}}(\xi'_{\tilde{\kappa}}(\gamma x), s'_{\tilde{\kappa}}(\gamma x)) \\ &= \frac{\gamma\, s'_\kappa(x)}{1+\sqrt{|\tilde{\kappa}|}\,\gamma\,\xi'_\kappa(x)} \\ &= \frac{\gamma\, s'_\kappa(x)}{1+\sqrt{|\kappa|}\,\xi'_\kappa(x)} \\ &= \gamma\,\rho_\kappa(\xi'_\kappa(x), s'_\kappa(x)) \\ &= \gamma\, F_\kappa(x). \end{aligned}$$

Dividing both sides by γ yields

$$F_\kappa(x) = \frac{1}{\gamma}\, F_{\kappa/\gamma^2}(\gamma x),$$

which completes the proof. □

Remark. The above scale-equivariance property does *not* hold if the curvature κ is kept fixed while scaling the input x. This is due to the non-homogeneous dependence on κ in both the inverse stereographic projection denominator and the curvature-dependent offset $\varphi(\kappa)$.

Table 12: Runtime and memory consumption comparison between MoS and Exp/Log geometric mappings. Results are reported for both forward and backward passes.

Method	Total Time (ms)	Phase	Avg Time (ms)	Avg Memory (MB)
MoS	**57.4**	Forward	**2.1**	**277.2**
		Backward	**2.1**	**324.0**
Exp/Log	135.0	Forward	2.6	565.6
		Backward	8.6	396.0

F EFFICIENCY

We evaluate the computational efficiency of the proposed MoS framework by comparing it against the standard exponential/logarithmic (Exp/Log) mapping. For each method, we report the total execution time, as well as the average runtime and memory consumption for both forward and backward passes. All measurements are conducted under identical hardware and batch-size settings to ensure fair comparison.

As shown in Table 12, MoS substantially reduces the overall conversion time compared to the Exp/Log mapping. In particular, MoS achieves a total runtime of 57.46 ms, less than half of the 135.08 ms required by Exp/Log.

In terms of memory usage, MoS demonstrates significantly improved efficiency during the forward pass, consuming approximately 277 MB, compared to 566 MB for Exp/Log. During the backward pass, MoS requires around 324 MB of memory, which is comparable to the baseline. Overall, these results indicate that MoS not only accelerates geometric transformations but also maintains competitive or lower memory consumption, especially in the forward pass, thereby offering superior computational efficiency.