

Riemannian Large Language Modeling

Shu Xu¹, Jiahong Liu², Wenhao Zhuang⁵, Jiabao Li³,
Yefan Wang⁴, Ming Sun⁵, Menglin Yang^{1,*}

¹ Hong Kong University of Science and Technology (Guangzhou);

² The Chinese University of Hong Kong; ³ Peking University;

⁴ Tsinghua University; ⁵ Kuaishou Technology

*Corresponding Author

Abstract

Current large language models perform next-token prediction under the implicit assumption that all tokens lie in a shared Euclidean embedding space, which treats every token uniformly. However, natural language contains rich structural relationships, including hierarchical, compositional, and cyclic patterns, which the flat geometry of Euclidean space cannot model well. To address this limitation, we propose rLLM (Riemannian Large Language Modeling), which introduces a token-adaptive positive-definite diagonal metric tensor into the LM head. This metric dynamically adjusts the local quadratic score for each hidden state and its context, allowing the model to better capture the relationships among tokens and prompts, with less than 0.01% additional parameters. We train rLLM from scratch across three different scales (2.8B to 5.0B parameters), covering the full pretraining, SFT, and RL pipeline. Extensive experiments show that rLLM improves Base CORE, a composite of 22 in-context learning tasks, by up to +2.5pp at the two smaller GPT scales and obtains higher ChatCORE at all three scales, with gains up to +3.0pp after RL. These results suggest that revisiting the geometric assumptions underlying token representations is a promising direction for building more expressive and capable language models. Code is available at <https://github.com/digailab/riemannianllm.git>.

1 Introduction

Large language models (LLMs) have achieved remarkable progress across a wide range of tasks, yet they rest on a strong and rarely questioned geometric assumption. Modern architectures such as GPT (Radford et al., 2019; Brown et al., 2020), LLaMA (Touvron et al., 2023), and Qwen (Qwen Team, 2025) perform next-token prediction under the implicit assumption that all tokens lie in a shared Euclidean embedding space, treating every direction uniformly. This assumption is inher-

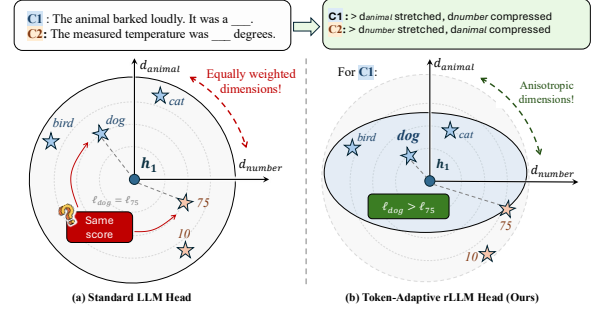


Figure 1: Geometric intuition. A standard LM head uses an isotropic geometry, while rLLM uses a token-adaptive metric that reweights semantic directions by context.

ited almost universally across model families and scales, despite the fact that the linguistic relationships these embeddings are meant to encode are far from geometrically uniform.

Natural language is geometrically diverse, imposing fundamentally different demands that a single Euclidean metric cannot simultaneously satisfy. On the linguistic side, analogical relations favor parallelogram-like structures (Mikolov et al., 2013), semantic taxonomies are better represented in hyperbolic space (Nickel and Kiela, 2017), and positional patterns can be modeled through rotational transformations (Su et al., 2024). On the empirical side, contextual embeddings across BERT, GPT-2, and modern LLMs are highly anisotropic, concentrating in narrow cones rather than spreading across the available space (Ethayarajh, 2019; Gao et al., 2019); hierarchical concepts are encoded through approximately orthogonal structures (Park et al., 2025); and representation spaces across large vision and language models become increasingly aligned as model capability grows (Huh et al., 2024). A recent position paper argues that this is a foundational issue, contending that moving beyond Euclidean geometry is a necessity rather than an optional enhancement for the next generation of

foundation models (He et al., 2025b).

Imposing a Euclidean metric on inherently non-Euclidean structure carries concrete costs. For hierarchical structures, Euclidean embeddings incur geometric distortion or require high dimensionality to preserve pairwise relationships (Nickel and Kiela, 2017; Sala et al., 2018). The expressiveness of softmax distributions defined over Euclidean inner products is also inherently bounded by the softmax bottleneck, forcing models to rely on ever-larger embedding dimensions to approximate the true conditional distribution (Yang et al., 2018). Together, these observations point to a clear research gap: the geometric foundation of the LM head has not kept pace with our understanding of how language is actually structured in representation space.

Riemannian geometry offers a compact way to relax this limitation. A positive-definite metric tensor reweights embedding directions and turns the fixed Euclidean inner product into a learned distance function. This view connects to geometric deep learning and prior work on Poincaré embeddings, hyperbolic networks, Riemannian optimization, and adaptive graph metrics (Bronstein et al., 2021; Nickel and Kiela, 2017; Ganea et al., 2018; Bonnabel, 2013; Lou et al., 2020; Wang et al., 2026), while existing language-model studies use broader manifold or hyperbolic constructions; our approach confines the geometric change to the LM head.

We propose rLLM (Riemannian Large Language Modeling), a language modeling framework that introduces a token-conditioned positive-definite diagonal metric in the LM head. The resulting local quadratic score adapts to each hidden state and its context. This adds less than 0.01% parameters relative to the base model, keeping rLLM readily compatible with existing pretraining infrastructure. Figure 1 gives the geometric intuition behind this token-adaptive prediction rule. We evaluate rLLM extensively across three model scales, the full pretraining, SFT, and RL pipeline, and diverse CORE and ChatCORE tasks. rLLM improves Base CORE at the two smaller GPT scales in the reported runs and obtains higher RL ChatCORE at all three scales, with absolute score gains up to 0.025 in Base CORE and 0.030 in ChatCORE after RL.

Our contributions can be summarized as follows. **(1) A head-level Riemannian language modeling framework at billion-parameter scale.** We intro-

duce a learnable local metric head and successfully train models with up to 5.0B parameters, achieving absolute score gains of 0.025 in Base CORE at 2.8B and 0.030 in ChatCORE at 3.8B over Euclidean baselines. **(2) A token-adaptive metric tensor for context-dependent geometry.** A lightweight module produces a per-token positive-definite metric that dynamically adapts to each token’s context, capturing geometric relationships inaccessible to a fixed Euclidean inner product. **(3) Improved multi-task robustness during post-training.** In the reported runs, rLLM maintains SpellingBee performance during RL while the Euclidean baseline shows a large drop, and rLLM achieves the highest ChatCORE among the evaluated heads at each scale.

2 Related Work

Geometry of language representations. Word embeddings exhibit linear regularities (Mikolov et al., 2013), while hyperbolic and spherical spaces provide useful inductive biases for hierarchical and directional structure (Nickel and Kiela, 2017; Meng et al., 2019). Recent analyses further characterize anisotropy, degeneration, and hierarchical concept geometry in contextual representations (Ethayarajh, 2019; Gao et al., 2019; Park et al., 2025). Recent surveys organize hyperbolic foundation models by how they modify embedding spaces and model operations (He et al., 2025c), with recent methods spanning hyperbolic fine-tuning, multimodal continual learning, and fully hyperbolic Transformers (Yang et al., 2025; Liu et al., 2026; Yang et al., 2024; He et al., 2025a). Our work complements these directions by retaining the Euclidean Transformer body and introducing token-conditioned geometry only in the output head.

Riemannian and mixed-curvature neural networks. Recent surveys summarize hyperbolic graph learning methods and applications (Yang et al., 2022b). Geometric learning has developed manifold-valued layers for graph convolution and recommendation (Chami et al., 2019; Yang et al., 2022a), product manifolds (Gu et al., 2019), and constant-curvature spaces (Bachmann et al., 2020). More recent methods combine curvature with spectral structure or adapt geometry to heterogeneous graph topology (Grover et al., 2025; Guo et al., 2025b; Wang et al., 2026). These approaches establish the value of flexible geometric inductive biases for graph representation learning. In con-

trast, rLLM targets next-token prediction at billion-parameter scale and learns a lightweight local metric conditioned on each token representation.

Representation rank and output expressiveness. The softmax bottleneck connects the rank of hidden representations to the family of output distributions a language model can express (Yang et al., 2018). Related studies document dimensional collapse and feature distortion in representation learning and fine-tuning (Jing et al., 2022; Kumar et al., 2022). We use effective rank as a diagnostic of the hidden-state spectrum and pair it with task-level evaluation across pretraining and post-training stages.

3 Method

We take a different approach from prior Riemannian deep learning methods, which typically map representations onto a curved manifold and replace standard operations with their manifold-valued counterparts such as exponential maps, parallel transport, and Riemannian optimizers (Nickel and Kiela, 2017; Ganea et al., 2018; Bonnabel, 2013; He et al., 2025b). At LLM scale, this adds substantial overhead and complicates training infrastructure. In rLLM, both the Transformer body and the next-token prediction objective remain unchanged. The geometric change is confined to the LM head. Because the metric is positive definite by construction while the trainable weights remain Euclidean parameters, standard optimizers are sufficient. Figure 2 summarizes this head-level replacement.

3.1 Preliminaries

Riemannian geometry and metric tensors. A Riemannian manifold is a smooth space in which each point is equipped with an inner product, called the *metric tensor*, that defines local notions of length, angle, and distance. Formally, a metric tensor at a point $\mathbf{x}$ is a symmetric positive-definite matrix $G(\mathbf{x}) \in \mathbb{R}^{d \times d}$, and the squared length of an infinitesimal displacement $d\mathbf{x}$ is given by $d\mathbf{x}^\top G(\mathbf{x}) d\mathbf{x}$. The Euclidean space is the special case $G(\mathbf{x}) \equiv I$, where every direction is treated identically and distance reduces to the familiar form $\|\mathbf{x} - \mathbf{y}\|^2 = (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})$. Choosing a non-identity metric reweights or rotates the coordinate axes, so that some directions become geometrically closer than others. When the metric depends on the hidden state, different contexts use different local metric tensors. This parameteriza-

tion does not by itself assert a particular curvature or a global manifold geometry.

3.2 A Distance-Based Reformulation of the LM Head

From inner product to distance. We begin by relating the standard inner-product LM head, $\ell_v = \mathbf{h}_t^\top \mathbf{w}_v$, to a distance-based score, following the distance-based classification view used in prototypical networks (Snell et al., 2017). Expanding the negative squared Euclidean distance between the hidden state $\mathbf{h}_t$ and the classifier weight $\mathbf{w}_v$ for token v yields

$$-\frac{1}{2} \|\mathbf{h}_t - \mathbf{w}_v\|^2 = \mathbf{h}_t^\top \mathbf{w}_v - \frac{1}{2} \|\mathbf{w}_v\|^2 - \frac{1}{2} \|\mathbf{h}_t\|^2. \quad (1)$$

For a fixed context t , the same hidden state $\mathbf{h}_t$ is used to score every candidate token v , so the $-\frac{1}{2} \|\mathbf{h}_t\|^2$ term adds the same scalar to every vocabulary logit. Since softmax is invariant to adding a shared offset to all logits, this term does not change the conditional distribution. The prototype term $-\frac{1}{2} \|\mathbf{w}_v\|^2$, however, varies with v and cannot be absorbed into a shared constant. Thus the standard bias-free inner-product head and the identity-distance head below are related but distinct:

$$s_v^{\text{Id}} = -\frac{1}{2} (\mathbf{h}_t - \mathbf{w}_v)^\top \mathbf{I} (\mathbf{h}_t - \mathbf{w}_v) + \text{const}, \quad (2)$$

with the identity matrix $\mathbf{I}$ acting as the metric tensor for this distance-based score. Sections 3.3 and 3.4 obtain rLLM by allowing this metric tensor to depart from the identity, first globally and then in a token-adaptive way.

3.3 Generalizing to a Riemannian Metric

The formulation in Section 3.2 writes a distance-based head as a Euclidean quadratic score with an implicit identity metric. We now generalize this metric to a learnable, positive-definite diagonal tensor, giving the head a context-independent direction reweighting.

From identity to learnable metric. We replace the identity matrix in Eq. (2) with a diagonal metric $\mathbf{G} = \text{diag}(\mathbf{g})$, where $\mathbf{g} \in \mathbb{R}_{>0}^d$, giving the Mahalanobis form (Xing et al., 2002; Weinberger and Saul, 2009)

$$\ell_v = -\frac{1}{2} (\mathbf{h}_t - \mathbf{w}_v)^\top \mathbf{G} (\mathbf{h}_t - \mathbf{w}_v). \quad (3)$$

Each component g_i scales the contribution of dimension i to the squared distance: directions with $g_i > 1$ are emphasized, while directions with

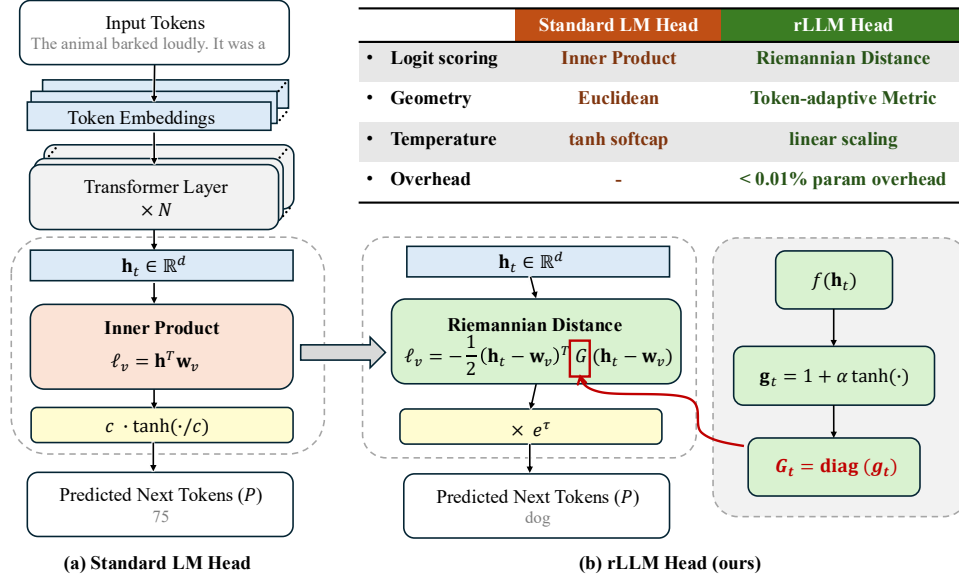


Figure 2: Method overview. rLLM keeps the Transformer body unchanged and replaces the output head with a local quadratic score using a token-adaptive diagonal metric. **(a)** The standard LM head computes logits via Euclidean inner products with tanh softcapping. **(b)** rLLM uses a token-conditioned local metric with a lightweight MLP $f(\mathbf{h}_t)$ producing a positive-definite diagonal tensor $\mathbf{G}_t = \text{diag}(\mathbf{g}_t)$, adding <0.01% parameters.

$g_i < 1$ are suppressed. Compared to the identity metric, this allows the LM head to weight coordinates non-uniformly and to capture direction-dependent similarity that the Euclidean inner product cannot express.

Why diagonal? We restrict $\mathbf{G}$ to be diagonal for two reasons. First, a diagonal metric already breaks the rotational symmetry of the Euclidean inner product and captures direction-dependent similarity. Second, a full $d \times d$ positive-definite matrix would add substantial parameter and computational overhead at the vocabulary scale, while a diagonal $\mathbf{G}$ adds only d parameters, which is negligible relative to the base model.

Coverage of common geometries. Although diagonal, this metric family is motivated by local tensor forms that occur in several standard geometries. In conformal coordinates, both the Poincaré ball and spherical models have metric tensors of the form $G(\mathbf{x}) = \lambda(\mathbf{x})^2 I$, and product mixed-curvature spaces yield block-wise conformal diagonal tensors. These examples motivate local diagonal or conformal reweightings; they do not mean that rLLM instantiates any global constant-curvature space. Appendix D gives the corresponding tensor forms.

Limitations of a global metric. A globally fixed $\mathbf{g}$ defines one shared coordinate reweighting across

the vocabulary. While more expressive than the Euclidean baseline, it forces every token to be predicted under the same reweighting, which we find limiting in practice (Section 6.3). Section 3.4 addresses this by allowing $\mathbf{G}$ to vary with the hidden state.

3.4 Token-Adaptive Riemannian Geometry

We now let the metric depend on the hidden state, so that different token contexts use different local metric tensors.

Token-adaptive metric. We define the per-token metric tensor as

$$\mathbf{g}_t = \mathbf{1} + \alpha \tanh(f(\mathbf{h}_t)), \quad \mathbf{g}_t \in [1-\alpha, 1+\alpha]^d, \quad (4)$$

where $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a two-layer MLP with bottleneck width $k = 64$ and a tanh output scaled by $\alpha = 0.5$, so that $g_{t,i} \in [0.5, 1.5]$ for every coordinate. The local tensor $\mathbf{G}_t = \text{diag}(\mathbf{g}_t)$ varies with the hidden state.

Logit scale. Distance logits still require a global scale to match the sharpness of the output distribution. Instead of tanh softcapping (Gemini Team, 2024), which saturates large distances nonlinearly, we use a learnable scalar temperature, as in Radford et al. (2021),

$$\ell_v \leftarrow e^\tau \ell_v, \quad \tau \text{ initialized to } 0. \quad (5)$$

This preserves the distance geometry because it rescales all logits linearly.

Substituting $\mathbf{G}_t$ into the Mahalanobis form of Eq. (3) and applying this temperature gives the final rLLM logit

$$\ell_v = e^{\tau} \left[(\mathbf{h}_t \odot \mathbf{g}_t)^\top \mathbf{w}_v - \frac{1}{2} \sum_{i=1}^d g_{t,i} w_{v,i}^2 - \bar{\ell} \right], \quad (6)$$

where $\bar{\ell}$ is the vocabulary mean logit, detached from the computation graph, subtracted for numerical stability. The hidden-state-dependent term $-\frac{1}{2} \sum_i g_{t,i} h_{t,i}^2$ is absorbed into $\bar{\ell}$ because it is constant across the vocabulary dimension and therefore does not affect the softmax distribution.

Parameter overhead. The adaptive module adds only 295K parameters, or 0.008% of a 3.8B model, making the overhead of moving from a Euclidean to a token-adaptive Riemannian head effectively negligible.

4 Experimental Setup

Model Architecture We build on nanochat (Karpathy, 2025), an end-to-end LLM training harness that uses a GPT-2-family architecture enhanced with modern components: Rotary Position Embeddings (Su et al., 2024), QK-normalization, ReLU² activation (Shazeer, 2020), grouped-query attention, Flash Attention (Dao et al., 2022), untied embeddings, and a sliding window attention pattern (three sliding-window layers followed by one full-context layer, tiled across depth). All hyperparameters derive from a single depth argument via muP-style scaling (Yang et al., 2021).

Model Scales We train at three scales using the model sizes and token budgets in Table 1, following the compute–data balancing principle of Hoffmann et al. (2022). The primary GPT baseline and rLLM runs use FP8 precision on 64 NVIDIA H100 GPUs across 8 nodes; precision exceptions for ablations and auxiliary studies are listed in Appendix A.

Training Pipeline and Evaluation All models follow a three-stage pipeline: pretraining on FineWeb-Edu (Penedo et al., 2024), supervised fine-tuning (SFT) on SmolTalk (Allal et al., 2025) augmented with MMLU and GSM8K training data, and reinforcement learning (RL) via a simplified

Depth	d_{model}	Heads	Params	Tokens	Steps
32	2048	16	2.82B	20.1B	9,601
36	2304	18	3.80B	28.4B	13,555
40	2560	20	4.99B	38.8B	18,481

Table 1: Model configurations. All models use head_dim=128, vocabulary size $\approx 50,000$, and sequence length 2048. Training uses FP8 precision with the Muon optimizer (Jordan et al., 2024) on 64 NVIDIA H100 GPUs.

GRPO variant (Shao et al., 2024; Guo et al., 2025a) on GSM8K (Cobbe et al., 2021) with token-level normalization (Yu et al., 2025). Training configuration details are provided in Appendix A.

We report two composite metrics (see Appendix B for full task lists and scoring details): **CORE** (primary pretrain evaluation): aggregates 22 in-context learning tasks across five categories (language understanding, world knowledge, commonsense reasoning, symbolic problem solving, and reading comprehension), with accuracy centered by random baselines, providing a comprehensive measure of pretrained model quality that is unaffected by post-training choices. **Chat-CORE** (post-fine-tuning evaluation): averages centered accuracies across six tasks, including ARC-Easy and ARC-Challenge (Clark et al., 2018), MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and SpellingBee, to measure how well pretrained representations translate into downstream capabilities. Together, these metrics capture both intrinsic model quality and practical task performance.

5 Results

5.1 Main Results

Figure 3 presents results comparing the Euclidean baseline against rLLM across three scales and the full training pipeline.

Improved pretrain quality. rLLM improves Base CORE, a composite of 22 in-context learning tasks, at two of three scales: +2.5pp at 2.8B (0.354 vs. 0.329) and +1.2pp at 3.8B (0.377 vs. 0.365). At 5.0B, Base CORE is comparable (−0.3pp). The pretrain improvement is more pronounced at smaller scales, where the baseline has greater room for geometric inefficiency. Val BPB is comparable: at 2.8B rLLM is slightly *better* (−0.006), while at 3.8B and 5.0B it is marginally higher (+0.003 and +0.001, respectively).

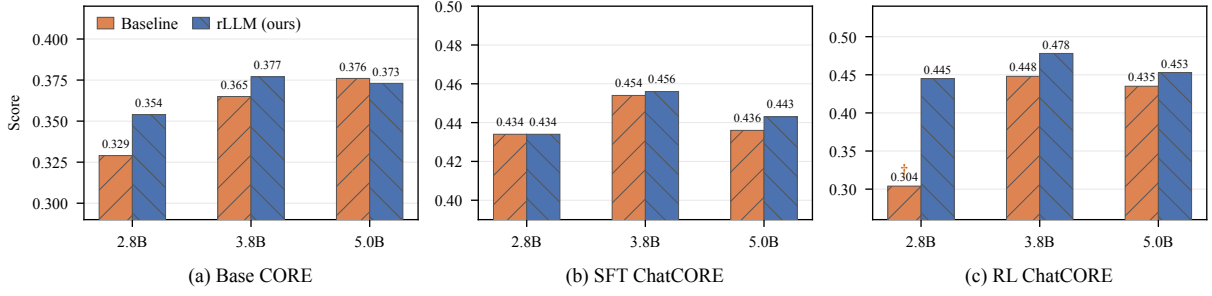


Figure 3: Performance comparison across three model scales and pipeline stages. Base CORE improves at 2.8B (+2.5pp) and 3.8B (+1.2pp); at 5.0B it is comparable. RL ChatCORE improves at every scale, with +3.0pp at 3.8B and +1.7pp at 5.0B. [†]Baseline 2.8B RL ChatCORE is anomalously low due to SpellingBee collapse.

Consistent post-training gains. Across scales, rLLM achieves the highest RL ChatCORE, with 0.445 at 2.8B, 0.478 at 3.8B (+3.0pp), and 0.453 at 5.0B (+1.7pp). The positive 5.0B post-training gain persists even though Base CORE is comparable, motivating our analysis of the hidden-state spectrum in Section 5.2. Within the main baseline comparison, SFT ChatCORE is tied at 2.8B and higher at 3.8B and 5.0B.

5.2 Representation Diversity: Effective Rank

We next examine how the Riemannian metric affects the representational structure of hidden states. We measure the effective rank (Roy and Vetterli, 2007) of last-layer hidden states $H \in \mathbb{R}^{n \times d}$ across 100 GSM8K test problems:

$$\text{erank}(H) = \exp\left(-\sum_i p_i \log p_i\right), \quad (7)$$

$$p_i = \frac{\sigma_i}{\sum_j \sigma_j}$$

where σ_i are the singular values of H .

Figure 4 reveals a consistent spectral gap: rLLM achieves approximately $5\times$ higher effective rank at the base stage and $3\times$ after RL. The baseline remains below 10 (out of $d=2048-2560$) at the base stage, utilizing less than 0.5% of available dimensions, while rLLM consistently reaches 44–51. This scale-consistent pattern indicates that the Riemannian metric enables more uniform information distribution across embedding dimensions. We analyze the mechanisms in Section 6.1.

5.3 Per-Task Analysis

Table 2 shows per-task RL scores at the two largest scales.

At 3.8B, the HumanEval gap is particularly notable: 0.140 vs. 0.085 (+5.5pp), despite Hu-

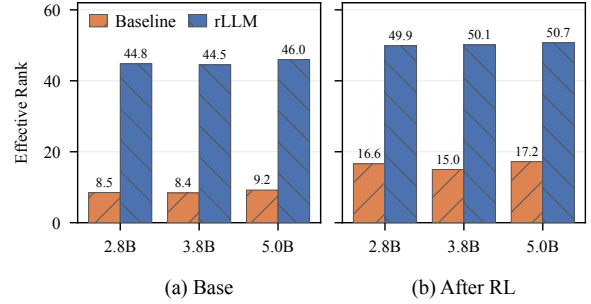


Figure 4: Effective rank of hidden states measured on 100 GSM8K test inputs across three scales. rLLM achieves approximately $5\times$ higher effective rank at the base stage and $3\times$ after RL, with a consistent gap across model sizes.

manEval being completely outside the RL training distribution (which targets only GSM8K). At 5.0B, a complementary pattern emerges: rLLM wins on four of six tasks (ARC-E/C, MMLU, HumanEval) but slightly trails on the RL target GSM8K (−0.6pp), suggesting that the geometric benefit primarily enhances out-of-distribution transfer rather than in-distribution optimization.

5.4 Multi-Task Robustness During RL

Figure 5 isolates the effect of RL by showing SFT→RL score changes across all six evaluation tasks at three scales.

At 2.8B (Figure 5a), baseline SpellingBee decreases from 0.992 to 0.188 (−80.5pp), while rLLM changes by −0.4pp. At 3.8B, rLLM shows a distinctive profile: no non-target task decreases by more than 0.2pp, while GSM8K increases by 10.8pp. HumanEval is completely outside the RL distribution, yet it increases by 1.2pp, making this the only configuration to show positive transfer. In contrast, baseline HumanEval and SpellingBee decrease by 5.5pp and 3.5pp, respectively. At 5.0B,

Scale	Method	ARC-E	ARC-C	MMLU	GSM8K	HumanEval	SpellingBee	ChatCORE
3.8B	Baseline	.770	.615	.427	.222	.085	.965	.448
	rLLM	.800	.645	.436	.228	.140	.992	.478
5.0B	Baseline	.746	.589	.415	.178	.104	.996	.435
	rLLM	.781	.606	.429	.172	.134	.988	.453

Table 2: Per-task RL scores at 3.8B and 5.0B. At 3.8B, rLLM outperforms the baseline on *all six tasks* and is the only configuration to achieve this. ARC-Easy reaches 0.800, the highest across all model scales.

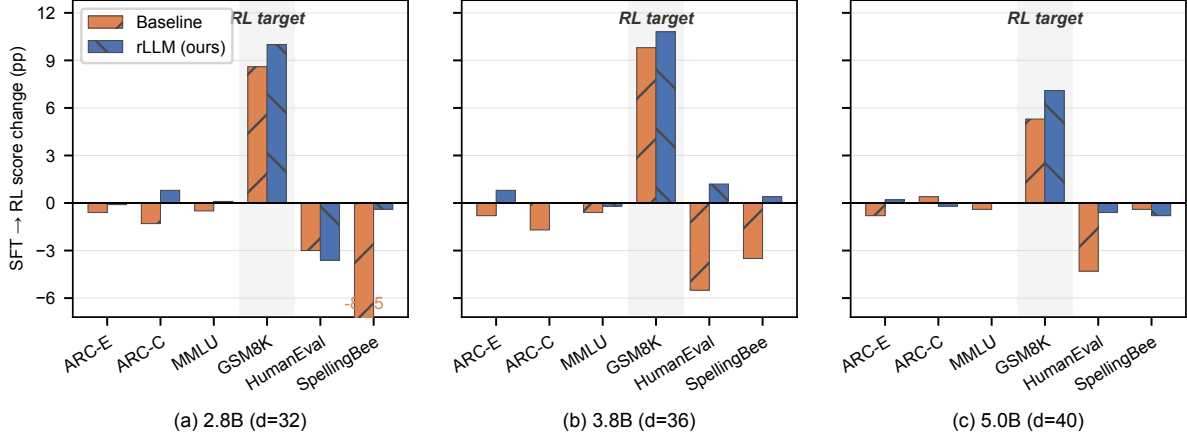


Figure 5: Per-task score changes from SFT to RL (pp) at 2.8B, 3.8B, and 5.0B. The shaded column marks GSM8K (RL target). At 2.8B, baseline SpellingBee decreases by -80.5pp (bar truncated), while rLLM changes by -0.4pp . At 3.8B, no rLLM non-target task decreases by more than 0.2pp . At 5.0B, rLLM has the smallest sum of negative non-target changes.

the same pattern holds: HumanEval decreases by 0.6pp under rLLM, compared with approximately 4.3pp under the baseline.

6 Analysis

6.1 Effective-Rank Observation

The effective-rank gap in Figure 4 is a consistent descriptive difference between the baseline and rLLM. One possible explanation is that the token-adaptive metric changes the per-dimension gradient scaling, but the current experiments do not establish this mechanism.

Token-adaptive gradient diversification. In rLLM, the logit computation involves the virtual hidden state $\tilde{\mathbf{h}} = \mathbf{h} \odot \mathbf{g}_t$, so the gradient for each dimension i of $\mathbf{h}$ is scaled by $g_{t,i}$. Because $\mathbf{g}_t$ varies across tokens, different token types may emphasize different dimensions in their gradient contributions: a math token may upweight numerical dimensions while a code token upweights syntactic ones. This could diversify per-dimension training signals relative to a fixed inner-product head, but it remains a hypothesis. We have no rank-matched intervention and no effective-rank measurements for the ablation heads, so the data cannot distinguish a use-

ful mechanism from a byproduct of the changed gradient path.

6.2 Effective Rank and Prior Analyses

Prior work relates representation rank to output expressiveness and representation anisotropy, which motivates measuring effective rank here. These relations provide context rather than a causal explanation for our task results.

Output expressiveness and token discriminability. Yang et al. (2018) show that the expressiveness of softmax output distributions is bounded by the rank of the logit matrix $L = HW^\top$, since $\text{rank}(L) \leq \text{rank}(H)$. When H concentrates in a low-dimensional subspace, the model may express a narrower family of output distributions across contexts. This is closely related to the representation anisotropy observed in standard language models (Ethayarajh, 2019; Gao et al., 2019): when hidden states concentrate in a low-dimensional cone, cosine similarities between unrelated tokens become artificially high, reducing discriminability. A higher measured effective rank is consistent with these prior motivations, but it does not by itself establish relaxed softmax bottleneck behavior or

Metric	Scale	Base	+ τ	Adpt.	Id.	Glob.	rLLM
Base CORE	2.8B	.329	.342	.335	.335	.341	.354
	3.8B	.365	.361	.379	.314	.337	.377
	5.0B	.376	.371	.369	.332	.332	.373
SFT ChatCORE	2.8B	.434	.414	.413	.417	.423	.434
	3.8B	.454	.428	.450	.406	.430	.456
	5.0B	.436	.434	.436	.411	.363	.443
RL ChatCORE	2.8B	.304 [†]	.422	.413	.427	.432	.445
	3.8B	.448	.435	.465	.417	.437	.478
	5.0B	.435	.442	.445	.433	.393	.453

Table 3: Ablation across three metrics and scales. Base is the standard baseline; + τ adds learnable e^τ scaling to the inner-product head; Adpt. is adaptive linear; Id. and Glob. are identity and global-diagonal distance heads. [†]Baseline 2.8B RL is anomalous due to SpellingBee collapse.

better task performance in our experiments. The 5.0B result provides a direct counterexample to treating rank as sufficient for a CORE gain.

6.3 Ablation Study

To isolate the contributions of individual components, we compare rLLM against five alternatives spanning the inner-product and distance-based formulations (Table 3). The first three methods use the standard inner-product $\ell_v = \mathbf{h}^\top \mathbf{w}_v$ with progressively more learnable components; the last three use the distance-based formulation $\ell_v = e^\tau [-d_G^2(\mathbf{h}, \mathbf{w}_v)]$ with different metric structures.

Table 3 shows that, in these single-run comparisons, rLLM achieves the highest RL ChatCORE at every scale. The adaptive-linear head is close on Base CORE at the larger scales (.379 vs. .377 at 3.8B and .369 vs. .373 at 5.0B), so the current ablation does not isolate the contribution of the distance term; its BF16 pretraining also differs from the FP8 rLLM runs. The global-diagonal head performs poorly at 5.0B (ChatCORE 0.363); gradient cancellation is a possible but untested explanation. Additional hyperparameter sensitivity analysis is provided in Appendix E.

6.4 Cross-Architecture Generalization

To test whether the benefits of rLLM are specific to the GPT-2 architecture or reflect a more general principle, we replicate the experiment on Qwen3 (Qwen Team, 2025), a modern architecture with distinct design choices including grouped-query attention with different head ratios, SwiGLU activations, and RMSNorm. We apply the same head-swap protocol: replacing only the output projection while keeping the transformer body un-

Metric	Depth	Baseline	rLLM
Base CORE	depth=32	.339	.348
	depth=36	.358	.367
	depth=40	.362	.370
SFT ChatCORE	depth=32	.462	.465
	depth=36	.478	.479
	depth=40	.500	.491
RL ChatCORE	depth=32	.479	.480
	depth=36	.500	.503
	depth=40	.514	.515

Table 4: Cross-architecture validation on Qwen3 at three depths (depth 32: 1.9B, depth 36: 2.6B, depth 40: 3.6B params; corresponding GPT-2 scales are 2.8B, 3.8B, 5.0B). rLLM improves Base CORE and RL ChatCORE at every depth.

changed.

Table 4 shows that rLLM transfers to Qwen3 across three depths, with absolute Base CORE gains of 0.008–0.009 and RL ChatCORE gains of 0.001–0.003 at every depth. The consistent direction across two architecturally distinct model families suggests that the benefit of adaptive Riemannian geometry in the output layer is not architecture-specific.

7 Conclusion

We proposed rLLM, a head-level Riemannian language modeling framework that replaces the fixed Euclidean inner-product LM head with a learnable, token-adaptive diagonal metric tensor. This simple modification yields an effective-rank gap of approximately $5\times$ at the base stage and $3\times$ after RL, a consistent descriptive spectral difference. In the reported runs, it also yields higher post-training scores and particularly strong multi-task robustness during RL fine-tuning, while the GPT Base CORE gain attenuates at larger scale.

Limitations

Model scale. Experiments cover scales up to 5.0B parameters across two architecture families. Whether the benefits persist at 10B+ with larger vocabularies, longer contexts, or multi-stage pre-training remains untested, though the negligible parameter overhead ($<0.01\%$) should scale favorably. We leave it for future work.

Ethics Statement

This work proposes a language-model training method using public corpora and standard benchmarks, without human-subject data collection or a new deployment scenario; however, models trained with this method may inherit biases, harmful content, privacy risks, and dual-use concerns from their training data, so dedicated safety, fairness, privacy, and misuse evaluations are required before any real-world deployment.

Acknowledgement

We thank the anonymous reviewers and the Area Chair for their constructive comments and valuable suggestions, which greatly improved the quality of this paper. This work was partially supported by the Guangdong Provincial Natural Science Foundation General Program (Grant No. 2026A1515012118) and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2025A1515140110). We also thank Kuaishou Technology for providing GPU resources.

References

- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. SmolLM2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.
- Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. 2020. Constant curvature graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 486–496. PMLR.
- Silvère Bonnabel. 2013. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.
- Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Øyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 55–65. Association for Computational Linguistics.
- Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, volume 31.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

- Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Karish Grover, Haiyang Yu, Xiang Song, Qi Zhu, Han Xie, Vassilis N. Ioannidis, and Christos Faloutsos. 2025. Spectro-Riemannian graph neural networks. In *International Conference on Learning Representations*.
- Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. 2019. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025a. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.
- Zihao Guo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, Min Zhou, Yisen Gao, and Jianxin Li. 2025b. GraphMoRE: Mitigating topological heterogeneity via mixture of Riemannian experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11754–11762.
- Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassioulas, Menglin Yang, and Rex Ying. 2025a. HELM: Hyperbolic large language models via mixture-of-curvature experts. In *Advances in Neural Information Processing Systems*, volume 38, pages 142604–142635. Curran Associates, Inc.
- Neil He, Jiahong Liu, Buze Zhang, Ngoc Bui, Ali Maatouk, Menglin Yang, Irwin King, Melanie Weber, and Rex Ying. 2025b. Position: Beyond Euclidean-foundation models should embrace non-Euclidean geometries. *arXiv preprint arXiv:2504.08896*.
- Neil He, Hiren Madhu, Ngoc Bui, Menglin Yang, and Rex Ying. 2025c. Hyperbolic deep learning for foundation models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD ’25, pages 6021–6031. ACM.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030.
- Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR.
- Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2022. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*.
- Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. [Muon: An optimizer for hidden layers in neural networks](#).
- Andrej Karpathy. 2025. [nanochat: The best ChatGPT that \\$100 can buy](#). GitHub repository.
- Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.
- Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, and 40 others. 2024. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 37, pages 14200–14282.
- Jiahong Liu, Ming Shen, Xiaohao Liu, Rex Ying, Menglin Yang, Tat-Seng Chua, and Irwin King. 2026. Hyperbolic multimodal continual learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR.
- Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. 2020. Differentiating through the Fréchet mean. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6393–6403. PMLR.
- Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *Advances in Neural Information Processing Systems*, volume 32.
- Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the*

- Association for Computational Linguistics: Human Language Technologies*, pages 746–751. Association for Computational Linguistics.
- Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, volume 30.
- Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*.
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben Alal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2024. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.
- Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). OpenAI technical report.
- Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610. IEEE.
- Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4460–4469. PMLR.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noam Shazeer. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.
- Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Xudong Wang, Chris Ding, Tongxin Li, and Jicong Fan. 2026. Adaptive Riemannian graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26606–26614.
- Kilian Q. Weinberger and Lawrence K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244.
- Eric P. Xing, Michael I. Jordan, Stuart J. Russell, and Andrew Y. Ng. 2002. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, volume 15.
- Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. 2021. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, pages 17084–17097.
- Menglin Yang, Ram B, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, and Rex Ying. 2025. Hyperbolic fine-tuning for large language models. In *Advances in Neural Information Processing Systems*, volume 38, pages 29613–29648. Curran Associates, Inc.
- Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. 2024. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’24, pages 3770–3781. ACM.
- Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. 2022a. HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization. In *Proceedings of the ACM Web Conference 2022*, WWW ’22, pages 2462–2471. ACM.
- Menglin Yang, Min Zhou, Tong Zhang, Jiahong Liu, Zhihao Li, Lujia Pan, Hui Xiong, and Irwin King. 2022b. Hyperbolic graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852*.
- Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*.

Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pages 113222–113244. Curran Associates, Inc.

A Training Configuration

The primary GPT baseline and rLLM models follow the same three-stage pipeline (pretrain $\rightarrow$ SFT $\rightarrow$ RL) and use FP8 precision on 64 NVIDIA H100 GPUs (8 nodes $\times$ 8 GPUs) with DDP. The adaptive-linear and learned-temperature ablations and the Qwen3 study use BF16; these precision differences are part of the reported protocol. The SFT mixture contains one epoch of SmolTalk, three repetitions of MMLU, four repetitions of GSM8K, 200K SimpleSpelling examples, and 80K SpellingBee examples. Hyperparameters are listed below.

Hyperparameter	Value
<i>Pretraining</i>	
Data	FineWeb-Edu
Training tokens	Scale-specific; see Table 1
Sequence length	2048
Optim. (matrices)	Muon (LR 0.02, WD 0.28)
Optim. (embeddings)	AdamW (LR 0.3)
Optim. (output proj.)	AdamW (LR 0.008)
LR schedule	40-step warmup, cosine to 5%
Precision	FP8 (Float8Linear)
<i>SFT</i>	
Data	SmolTalk (1ep) + MMLU (3ep) + GSM8K (4ep)
Optimizer	Same as pretrain
<i>RL (GRPO on GSM8K)</i>	
Algorithm	On-policy, no KL penalty
Normalization	Token-level (DAPO-style)
Advantages	Mean-centered (no std div.)
Examples per step	32
Samples per example	16
Temperature / top- k	1.0 / 50

Table 5: Training configuration for the primary GPT baseline and rLLM runs. Precision exceptions for ablations and Qwen3 are described in the text.

B Evaluation Metrics and Benchmarks

B.1 CORE (Base Model Evaluation)

CORE evaluates pretrained models via in-context learning across 22 tasks from the DCLM suite (Li et al., 2024). For each task i with raw accuracy a_i and random baseline b_i , centered accuracy is $\hat{a}_i = (a_i - b_i) / (1 - b_i)$, mapping random to 0 and

perfect to 1. CORE is the mean of all 22 centered accuracies.

Tasks use three protocols: **multiple choice** (lowest-loss option), **schema** (lower-loss context for coreference), and **language modeling** (free-form generation, exact match). Table 6 lists all 22 tasks.

Category	Task	Shots	Rand.
Language Underst.	hellaswag_zeroshot	0	25.0
	lambada_openai	0	0.0
	hellaswag	10	25.0
	winograd	0	50.0
	winogrande	0	50.0
	bigbench_lang_id	10	9.1
World Knowledge	jeopardy	10	0.0
	bigbench_qa_wikidata	10	0.0
	arc_easy	10	25.0
	arc_challenge	10	25.0
Common-sense	copa	0	50.0
	commonsense_qa	10	20.0
	pqa	10	50.0
	openbook_qa	0	25.0
Symbolic Prob. Solv.	bigbench_dyck_languages	10	0.0
	agi_eval_lsar_ar	3	20.0
	bigbench_cs_algorithms	10	0.0
	bigbench_operators	10	0.0
	bigbench_repeat_copy	10	0.0
Reading Compr.	squad	10	0.0
	coqa	0	0.0
	boolq	10	62.0

Table 6: The 22 CORE tasks. Rand. = random-chance accuracy (%).

B.2 ChatCORE (Post-Fine-Tuning Evaluation)

ChatCORE averages centered accuracies across six tasks: ARC-Easy, ARC-Challenge, and MMLU (categorical: lowest-loss option); GSM8K, HumanEval, and SpellingBee (generative: parsed free-form output).

Task	Random	Eval Type	Source
ARC-Easy	0.25	categorical	(Clark et al., 2018)
ARC-Challenge	0.25	categorical	(Clark et al., 2018)
MMLU	0.25	categorical	(Hendrycks et al., 2021)
GSM8K	0.00	generative	(Cobbe et al., 2021)
HumanEval	0.00	generative	(Chen et al., 2021)
SpellingBee	0.00	generative	procedural

Table 7: The six ChatCORE tasks. Random = chance-level baseline.

C Three-Seed Robustness Details

Table 8 gives the complete 2.8B seed study. SpellingBee uses 256 fixed test items and greedy decoding, so each score changes in steps of $1/256$. The reported mean and standard deviation are computed over the three paired initialization seeds; they are descriptive sample statistics, not confidence intervals or estimates of a population failure probability.

D Metric Tensor Coverage

The positive diagonal family used by rLLM is motivated by local tensor forms that occur in standard geometries. The Euclidean case is $G(\mathbf{x}) = I$. In conformal coordinates, the Poincaré ball and spherical models use tensors proportional to the identity, and products of conformal factors yield block-diagonal tensors. These expressions show that diagonal or conformal local reweightings are geometrically meaningful; they do not imply that the bounded field learned by rLLM instantiates any of these global constant-curvature spaces. rLLM neither matches their conformal factors nor integrates its metric along paths, and no curvature is inferred from this parameterization.

E Hyperparameter Sensitivity

All token-adaptive variants initialize the output layer of f to zero, so that $\mathbf{g}_t \equiv \mathbf{1}$ at the start of training. We compare design choices at 2.8B while holding the distance-based formulation constant (Table 9).

Variant	CORE	SFT	RL
tanh, $k=64$, $\alpha=0.5$ (ours)	.354	.434	.445
softplus, $k=64$ (unbounded)	.354	.436	.439
tanh, $k=512$ (default $d/4$)	.356	.432	.437
tanh, $k=64$, $\alpha=0.6$	.344	.413	.407
residual tanh, $k=64$	.352	.414	.425
asymmetric (1.5/0.3)	.336	.412	.428

Table 9: Hyperparameter sensitivity at 2.8B. Bounded tanh with $\alpha=0.5$ achieves the best RL ChatCORE.

Bounded metric in this study. $\alpha = 0.5$ ($\mathbf{g}_t \in [0.5, 1.5]$) obtains higher RL ChatCORE than $\alpha = 0.6$ in this 2.8B run (0.445 vs. 0.407). The unbounded softplus variant is competitive (0.439). These are single-run sensitivity results and do not establish that boundedness is universally essential.

Compact bottleneck suffices. A wider bottleneck ($k=512$) obtains lower RL performance

(0.437 vs. 0.445) in this run. The result is consistent with, but does not prove, that a compact token-adaptive module is sufficient.

F Additional Quantitative Results

Pretraining validation loss. Table 10 reports final validation bits-per-byte (BPB). The distance-based head has comparable validation BPB in these runs: at 2.8B, rLLM is lower than the baseline, while at 3.8B and 5.0B it is marginally higher by 0.003 and 0.001. These single-run values do not establish a general BPB advantage or disadvantage.

Method	2.8B	3.8B	5.0B
Baseline	0.6602	0.6403	0.6401
rLLM	0.6542	0.6430	0.6412
Δ (rLLM – Baseline)	−0.006	+0.003	+0.001

Table 10: Final validation BPB. Lower is better.

SFT→RL score changes. Table 11 decomposes the RL stage effect per task. At 3.8B, the reported rLLM run increases GSM8K by 0.108 and decreases no non-target task by more than 0.002; HumanEval increases by 0.012. These are single-run observations.

Scale	Method	Δ ARC-E	Δ ARC-C	Δ MMLU	Δ GSM8K	Δ HumanEval	Δ Spell.
3.8B	Baseline	−0.008	−0.017	−0.006	+0.098	−0.055	−0.035
	rLLM	+0.008	+0.000	−0.002	+0.108	+0.012	+0.004
5.0B	Baseline	−0.008	+0.004	−0.004	+0.053	−0.043	−0.004
	rLLM	+0.002	−0.002	+0.000	+0.071	−0.006	−0.008

Table 11: SFT→RL absolute score changes computed from full-precision task scores. Positive values indicate that RL improved the task.

Full per-task results. Table 12 presents complete per-task scores at all three scales including all distance-head variants.

Seed	Method	Base CORE	RL ChatCORE	ChatCORE _{-SB}	Spell. SFT	Spell. RL
10	Baseline	.331	.310	.326	.996	.230
10	rLLM	.352	.442	.332	.996	.992
20	Baseline	.335	.313	.324	.992	.258
20	rLLM	.361	.453	.344	.996	.996
42	Baseline	.329	.304	.327	.992	.188
42	rLLM	.354	.445	.336	.996	.992
Paired absolute Δ mean $\pm$ SD		+0.0240 $\pm$ 0.0026	+0.1377 $\pm$ 0.0049	+0.0116 $\pm$ 0.0077	–	–

Table 8: Complete 2.8B seed study. ChatCORE_{-SB} removes SpellingBee from the six-task average. Summary statistics are computed from full-precision scores before row-level rounding. The baseline SpellingBee SFT→RL absolute changes are -0.766 , -0.734 , and -0.805 ; the corresponding rLLM changes are -0.004 , 0.000 , and -0.004 .

Method	ARC-E	ARC-C	MMLU	GSM8K	HumanEval	SpellingBee	ChatCORE
<i>SFT scores at 3.8B (d=36)</i>							
Baseline	.778	.632	.433	.124	.140	1.000	.454
Dist. (identity)	.729	.564	.395	.095	.092	1.000	.406
Dist. (global diag)	.726	.603	.404	.106	.165	.996	.430
Adaptive (softplus)	.777	.630	.418	.132	.122	.996	.447
rLLM	.792	.645	.438	.120	.128	.988	.456
<i>RL scores at 3.8B (d=36)</i>							
Baseline	.770	.615	.427	.222	.085	.965	.448
Dist. (identity)	.734	.561	.402	.208	.067	.961	.417
Dist. (global diag)	.729	.594	.409	.216	.104	.992	.437
Adaptive (softplus)	.772	.625	.421	.204	.067	.996	.449
rLLM	.800	.645	.436	.228	.140	.992	.478
<i>SFT scores at 5.0B (d=40)</i>							
Baseline	.754	.585	.419	.125	.146	1.000	.436
Dist. (identity)	.731	.583	.399	.100	.092	.988	.411
Dist. (global diag)	.684	.515	.388	.046	.067	.949	.363
rLLM	.779	.608	.429	.102	.140	.996	.443
<i>RL scores at 5.0B (d=40)</i>							
Baseline	.746	.589	.415	.178	.104	.996	.435
Dist. (identity)	.744	.599	.406	.212	.067	.988	.433
Dist. (global diag)	.682	.526	.392	.223	.024	.977	.393
rLLM	.781	.606	.429	.172	.134	.988	.453
<i>SFT scores at 2.8B (d=32)</i>							
Baseline	.748	.601	.418	.106	.152	.992	.434
Dist. (identity)	.738	.585	.410	.083	.110	1.000	.417
Dist. (global diag)	.731	.589	.409	.093	.140	1.000	.423
Adaptive (softplus)	.754	.605	.422	.099	.146	.996	.436
rLLM	.766	.606	.415	.083	.140	.996	.434
<i>RL scores at 2.8B (d=32)</i>							
Baseline	.742	.588	.413	.192	.122	.188 [†]	.304 [†]
Dist. (identity)	.734	.589	.409	.202	.055	.996	.427
Dist. (global diag)	.732	.585	.410	.191	.104	.996	.432
Adaptive (softplus)	.754	.597	.420	.196	.085	.992	.439
rLLM	.765	.614	.416	.183	.104	.992	.445

Table 12: Full per-task scores at all three scales. ChatCORE is computed from full-precision task scores before the displayed task values are rounded. [†]Baseline SpellingBee decreased from 0.992 to 0.188 during RL at 2.8B.