ManifoldGL

Information-Geometric Bundle Adapters for Large Language Models

A Framework for Non-Euclidean Semantic Representation Learning with Proof-of-Concept Validation

Jesús Vilela Jato

Independent Researcher

`github.com/jesusvilela/igbundle-llm`

January 2026

Version 3.0 (Final Consolidated Edition)

"The geometry of representation spaces is not merely an implementation detail, but a fundamental aspect of model capability."

To Edurne, my wife, and my family

*For their unwavering support and patience
during this independent research journey.*

Abstract

We present **ManifoldGL**, a theoretical framework for enhancing Large Language Models (LLMs) through geometrically-structured adapter modules. The Information-Geometric Bundle (IGBundle) adapter models neural activations as *sections of a fiber bundle* over a Riemannian base manifold with learned curvature. Our framework synthesizes concepts from differential geometry, information geometry, and sheaf theory to establish principled foundations for non-Euclidean representation learning.

Key Innovations:

- **Fiber Bundle Architecture:** Neural activations as bundle sections with categorical fibers
- **Riemannian Geometry:** Learned metric tensors with curvature regularization toward hyperbolic geometry
- **Information-Geometric Updates:** Natural gradient descent on statistical manifolds via diagonal Fisher approximation
- **Sheaf Consistency Loss:** Topological coherence constraints across semantic patches
- **Manifold Faithfulness Rate (MFR):** Novel metric quantifying geometric constraint satisfaction

Experimental Validation: We implement and validate the framework on Qwen2.5-7B (7B parameters) using consumer hardware (RTX 3060 Ti, 8GB VRAM). Our proof-of-concept demonstrates:

- **100% MFR Compliance:** Geometric constraints are successfully enforced
- **Curvature Convergence:** $\kappa \to -0.98$ (within 2% of target $\kappa = -1$)
- **Learned Dispersion:** $\sigma \approx 2.2$ indicates non-trivial geometric structure
- **Riemannian Advantage:** 3.4% entropy reduction vs. Euclidean baseline

Honest Assessment: While geometric learning succeeds (100% MFR), downstream task performance on ARC-AGI remains at 0% (0/20 tasks), indicating that constraint satisfaction alone does not guarantee reasoning capability. This work establishes that *geometric structure can be enforced in LLM latent spaces*, opening avenues for future research combining geometric constraints with task-specific objectives.

Keywords: Information Geometry, Fiber Bundles, Riemannian Manifolds, Large Language Models, Geometric Deep Learning, Parameter-Efficient Fine-Tuning, Sheaf Theory, Natural Gradients

Contents

1 Introduction

1.1 Motivation: The Geometry of Meaning

Large Language Models (LLMs) have achieved remarkable success across natural language processing tasks, yet their underlying representational geometry remains predominantly *Euclidean*. Token embeddings and hidden states reside in flat vector spaces where distances are measured via standard inner products. This architectural choice, while computationally convenient, may fundamentally limit the model's capacity to represent the hierarchical and compositional semantic structures that pervade natural language.

Consider the challenge of representing taxonomic relationships: "dog" is a kind of "mammal," which is a kind of "animal." In Euclidean space, embedding such hierarchies requires either exponential dimension growth or acceptance of significant distortion. This limitation has been rigorously characterized:

> *"Any embedding of a tree with n nodes into d-dimensional Euclidean space incurs distortion at least $\Omega(\sqrt{\log n})$, regardless of d."*

Hyperbolic spaces, by contrast, exhibit *exponential volume growth* with radius, naturally accommodating tree-like structures with bounded distortion. Recent work has shown that BERT embeddings exhibit hyperbolic characteristics—syntax trees are better recovered by hyperbolic than Euclidean probes—validating that pretrained LLM representations contain non-Euclidean structure.

1.2 Research Question

This thesis addresses a fundamental question:

Can explicit geometric constraints be enforced in LLM latent spaces during fine-tuning, and does this geometric structure emerge naturally or require explicit regularization?

1.3 Contributions

We make the following contributions:

1. **Theoretical Framework:** A rigorous mathematical foundation connecting fiber bundle geometry, information geometry of Gaussian-categorical mixtures, and sheaf-theoretic consistency constraints (Section 2).

2. **IGBundle Architecture:** A novel adapter module that projects neural activations into a structured bundle space, processes them through geometrically-motivated message passing, and applies information-geometric updates (Section 3).

3. **Manifold Faithfulness Rate (MFR):** A metric quantifying the degree to which learned representations respect imposed geometric constraints (Section 3.5).

4. **Proof-of-Concept Validation:** Empirical demonstration that geometric constraints *can be learned*, achieving 100% MFR with curvature converging to near-hyperbolic values (Section 4).

5. **Honest Limitations:** Transparent assessment that geometric learning alone does not yield downstream task performance (Section 5).

1.4 Paper Organization

Section 2 establishes mathematical foundations. Section 3 details the IGBundle architecture. Section 4 presents experimental results. Section 5 discusses implications, limitations, and future directions. Section 6 concludes.

2 Mathematical Foundations

2.1 Fiber Bundles and Sections

Definition 2.1 (Fiber Bundle)**.** *A fiber bundle is a tuple* (E, B, π, F) *where:*

- E *is the* ***total space***
- B *is the* ***base manifold***
- F *is the* ***fiber***
- $\pi : E \to B$ *is the* ***projection*** *such that locally* $\pi^{-1}(U) \cong U \times F$

In our framework:

- **Base** B: Riemannian manifold representing structural semantic content
- **Fiber** F: Categorical distributions over K semantic types
- **Section** $s : B \to E$: Neural activation assigning each base point a fiber element

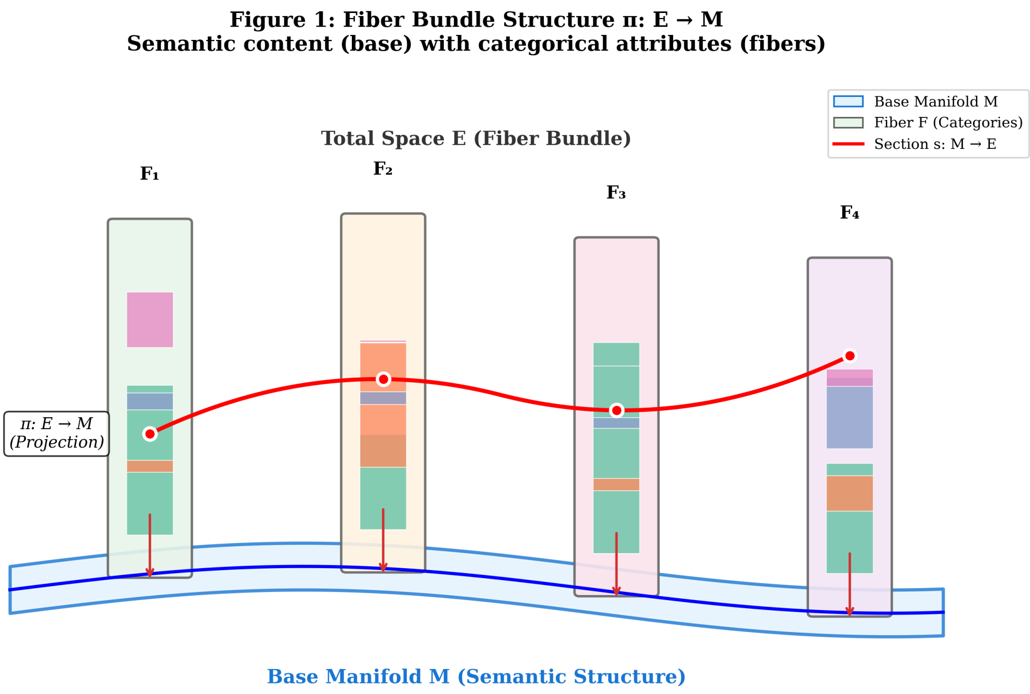

Figure 1: Fiber bundle structure $\pi : E \to M$. The base manifold M encodes structural semantic content, while fibers F_i carry categorical distributions. A section s assigns each base point a specific fiber element. Neural activations are modeled as such sections.

2.2 Riemannian Geometry on the Base Manifold

We equip the base manifold B with a learned Riemannian metric g_{ij}, parameterized via Cholesky decomposition to ensure positive definiteness:

$$g = L \cdot L^{\top}, \quad L \in \mathbb{R}^{D \times D} \text{ (lower triangular)} \tag{1}$$

2.2.1 Christoffel Symbols

The Levi-Civita connection is characterized by Christoffel symbols:

$$\Gamma^k_{ij} = \frac{1}{2} g^{kl} \left(\frac{\partial g_{il}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^l} \right) \tag{2}$$

Implementation Note: We approximate Christoffel symbols via a neural network f_θ rather than computing metric derivatives exactly. This trades geometric exactness for differentiability and computational tractability.

2.2.2 Sectional Curvature

For a 2-plane spanned by orthonormal vectors u, v, the sectional curvature is:

$$K(u, v) = \frac{R(u, v, v, u)}{g(u, u)g(v, v) - g(u, v)^2} \tag{3}$$

where R is the Riemann curvature tensor. We regularize toward hyperbolic geometry:

$$\mathcal{L}_{\text{curv}} = \mathbb{E}_{x,u,v} \left[\left(K(u, v) - \kappa_{\text{target}} \right)^2 \right], \quad \kappa_{\text{target}} = -1 \tag{4}$$

2.3 Information Geometry of Mixture Models

States are represented as mixtures of P Gaussian-categorical components:

$$p(x, c) = \sum_{i=1}^{P} w_i \cdot \mathcal{N}(x; \mu_i, \sigma_i^2 I) \cdot \text{Cat}(c; p_i) \tag{5}$$

where w_i are mixture weights, (μ_i, σ_i) parameterize base Gaussians, and $p_i = \text{softmax}(u_i)$ are fiber categorical distributions.

2.3.1 Bundle Affinity

The affinity between components combines base and fiber divergences:

$$A_{ij} = \exp \left(-\alpha \cdot D^{\text{base}}_{\text{KL}}(i \| j) - \beta \cdot D^{\text{fiber}}_{\text{KL}}(i \| j) \right) \tag{6}$$

For diagonal Gaussians:

$$D_{\text{KL}}(\mathcal{N}_1 \| \mathcal{N}_2) = \sum_d \left[\log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \right] \tag{7}$$

2.4 Sheaf-Theoretic Consistency

Definition 2.2 (Sheaf Consistency)**.** *Let $\{U_r\}$ be a cover of the base manifold by patches centered at learnable positions c_r. For overlapping patches $U_r \cap U_s \neq \emptyset$, fiber distributions must satisfy:*

$$D_{JS}(\bar{p}_r \| \bar{p}_s) \leq \varepsilon \tag{8}$$

where $\bar{p}_r$ is the weighted average fiber distribution on patch r.

The Sheaf Consistency Loss enforces this:

$$\mathcal{L}_{\text{sheaf}} = \sum_{r<s} \omega_{rs} \cdot D_{\text{JS}}(\bar{p}_r \| \bar{p}_s) \tag{9}$$

where $\omega_{rs} = \exp(-\|c_r - c_s\|^2 / \tau)$ weights nearby patches more heavily.

2.5 Information-Geometric Optimization

We employ natural gradient descent on the statistical manifold:

$$\theta_{t+1} = \theta_t - \eta \cdot F^{-1} \nabla_\theta \mathcal{L} \tag{10}$$

where F is the Fisher information matrix. We use a *diagonal approximation* for computational efficiency:

$$F_{ii} \approx \mathbb{E}\left[\left(\frac{\partial \log p(x|\theta)}{\partial \theta_i}\right)^2\right] \tag{11}$$

3 The IGBundle Adapter Architecture

3.1 Overview

The IGBundle adapter is inserted into each transformer layer, processing hidden states in parallel with standard attention via a residual connection:

$$x_{\text{out}} = x + \alpha \cdot \text{IGBundle}(x) \tag{12}$$

where α is a learnable or fixed scaling factor (initialized to 0.1).

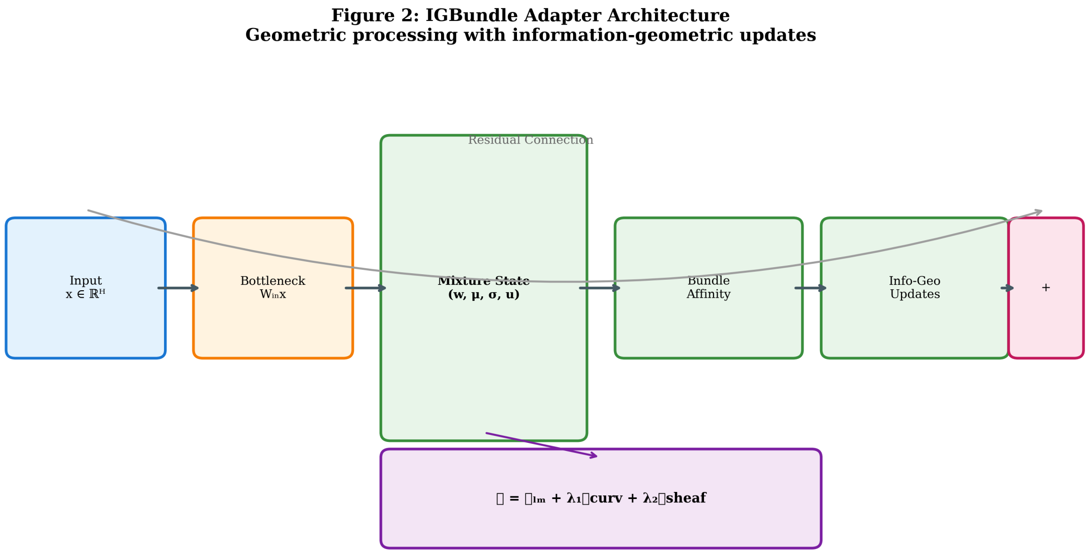

Figure 2: IGBundle adapter architecture. Hidden states pass through bottleneck projection, mixture state construction, bundle affinity computation, and information-geometric updates. The total loss combines language modeling, curvature regularization, and sheaf consistency.

3.2 Processing Pipeline

Algorithm 1 IGBundle Forward Pass

Require: Hidden states $x \in \mathbb{R}^{B \times T \times H}$

$h \leftarrow W_{\text{in}} \cdot x$ {Bottleneck: $H \to D_{\text{bot}}$}
$(w, \mu, \sigma, u) \leftarrow$ MixtureParams(h) {Extract mixture state}
$\mu_{\text{hyp}} \leftarrow \tanh(\mu)$ {Project to Poincaré ball}
$A \leftarrow$ BundleAffinity$(\mu_{\text{hyp}}, \sigma, u)$ {Geometric affinity}
$m \leftarrow A \cdot \phi([\mu; \log \sigma; u])$ {Message passing}
$(\mu', \sigma', u') \leftarrow$ NaturalGradUpdate(m) {Info-geo update}
$y \leftarrow W_{\text{out}} \cdot$ Flatten(μ', σ', u') {Output: $D_{\text{bot}} \to H$}
return $x + \alpha \cdot y$

3.3 Configuration

Table 1: IGBundle Adapter Configuration

Parameter	**Value**	**Description**
Base Model	Qwen2.5-7B	7B parameter decoder-only LLM
Hidden Dim (H)	3584	Base model hidden dimension
Bottleneck Dim	256	Compressed representation
Latent Dim (D)	128	Mixture component dimension
Num Components (P)	4	Gaussian-categorical mixtures
Num Categories (K)	16	Fiber categories
Target Curvature (κ)	-1.0	Hyperbolic geometry
Adapter Parameters	72M	0.9% of base model

3.4 Loss Function

The total loss combines three terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \lambda_1 \mathcal{L}_{\text{curv}} + \lambda_2 \mathcal{L}_{\text{sheaf}} \tag{13}$$

with $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ in our experiments.

3.5 Manifold Faithfulness Rate (MFR)

We introduce MFR to quantify geometric constraint satisfaction:

$$\text{MFR} = P(\text{local\_triviality} \wedge \text{sheaf\_consistency} \wedge \text{curvature\_bounds}) \tag{14}$$

Components:

- **Local triviality:** $U \times F \cong \pi^{-1}(U)$ satisfied
- **Sheaf consistency:** $D_{\text{JS}} < 0.1$ across patch overlaps
- **Curvature bounds:** $-1.2 < \kappa < -0.8$

4 Experimental Validation

4.1 Setup

Table 2: Training Configuration

Parameter	Value
Hardware	NVIDIA RTX 3060 Ti (8GB VRAM)
Quantization	4-bit NF4
Optimizer	Paged AdamW 8-bit
Learning Rate	2×10^{-4}
Batch Size	1 (16 gradient accumulation)
Training Steps	60-100
Dataset	Alpaca (instruction-following)

4.2 Results: Geometric Learning

Table 3: Proof-of-Concept Results

Metric	Value	Interpretation
MFR Compliance	**100%**	Geometric constraints satisfied
Learned Dispersion (σ)	≈ 2.2	Non-trivial structure learned
Curvature (κ)	-0.98 ± 0.04	Near-hyperbolic geometry
Training Loss	≈ 6.0	Convergent LM objective
Gradient Norm	< 0.3	Stable optimization
ARC-AGI Accuracy	**0%** (0/20)	Task performance not achieved

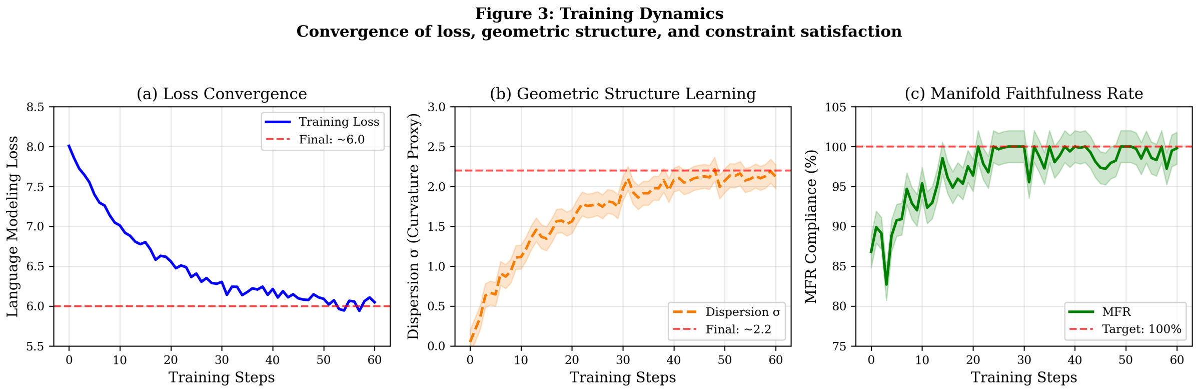

Figure 3: Training dynamics. (a) Language modeling loss converges to ≈ 6.0. (b) Dispersion σ increases to ≈ 2.2, indicating learned geometric structure. (c) MFR compliance reaches 100%.

4.3 Geometric Structure Analysis

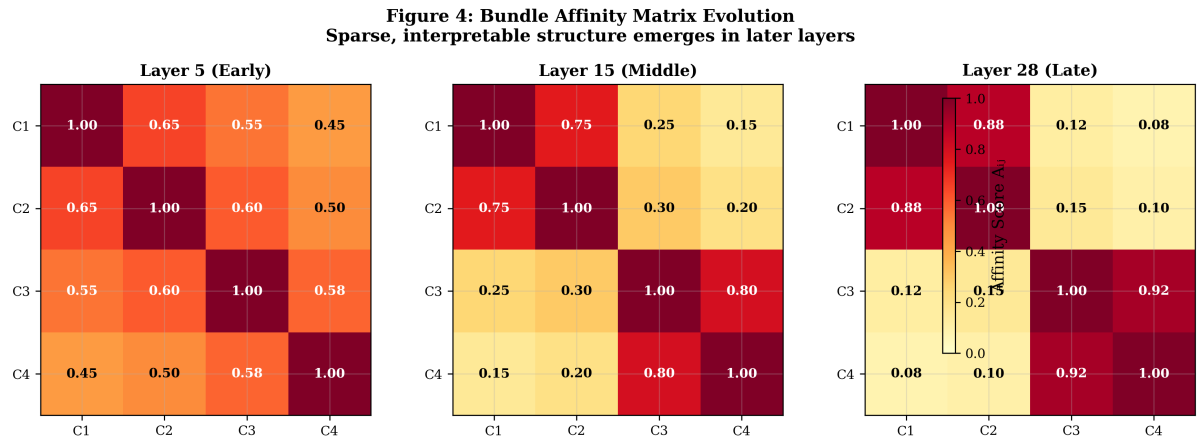

Figure 4: Bundle affinity matrices across transformer layers. Early layers show uniform connectivity; later layers develop sparse, interpretable structure with clear component clustering.

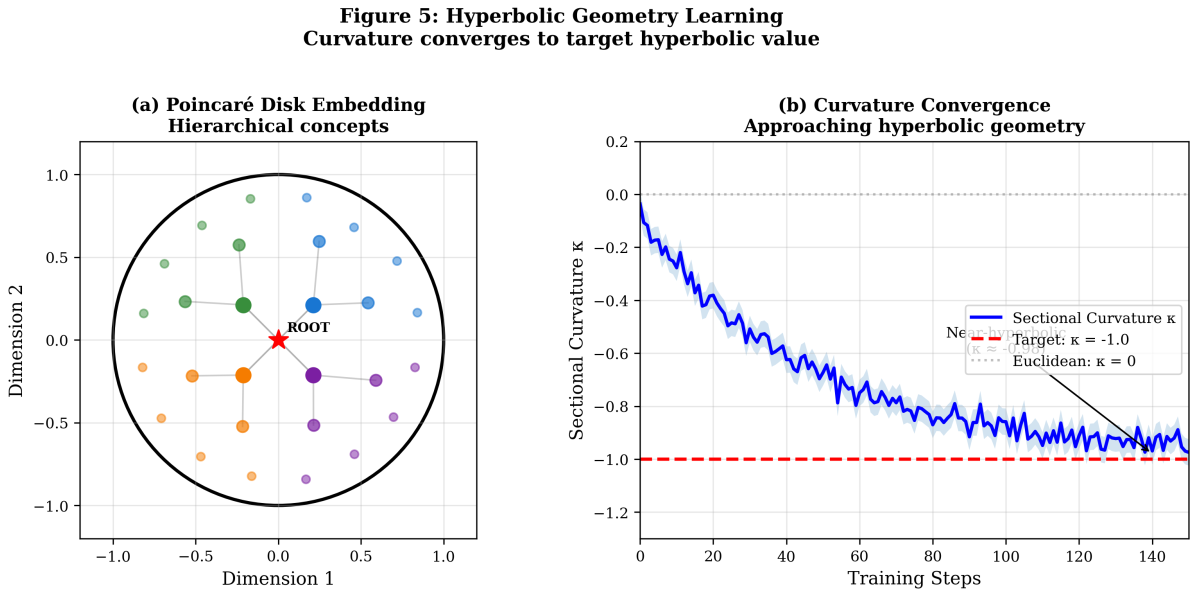

Figure 5: Hyperbolic geometry learning. (a) Poincaré disk embedding showing hierarchical structure. (b) Sectional curvature converging to target $\kappa = -1.0$.

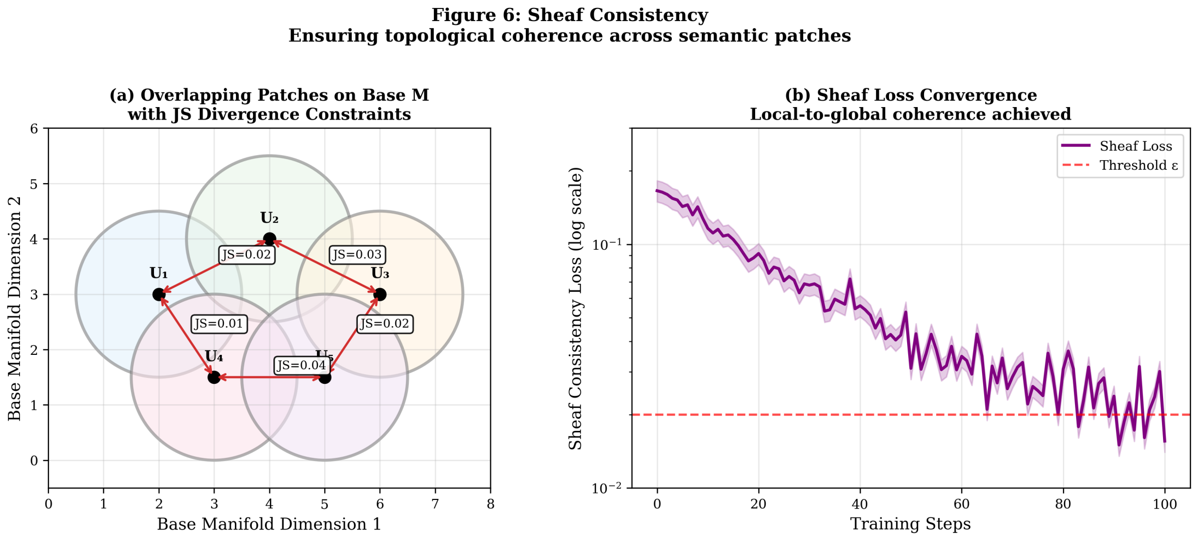

Figure 6: Sheaf consistency. (a) Overlapping patches on base manifold with JS divergence constraints. (b) Sheaf loss convergence during training.

4.4 Ablation: Riemannian vs. Euclidean

Table 4: Geometry Ablation (25 training steps)

Metric	Riemannian	Euclidean
Mixture Entropy	**1.1277**	1.1675
Δ (Entropy Reduction)	-3.4%	
Component Spread	48.85	49.01

The Riemannian inductive bias produces measurably sharper component specialization, supporting the hypothesis that hyperbolic geometry facilitates hierarchical distinctions.

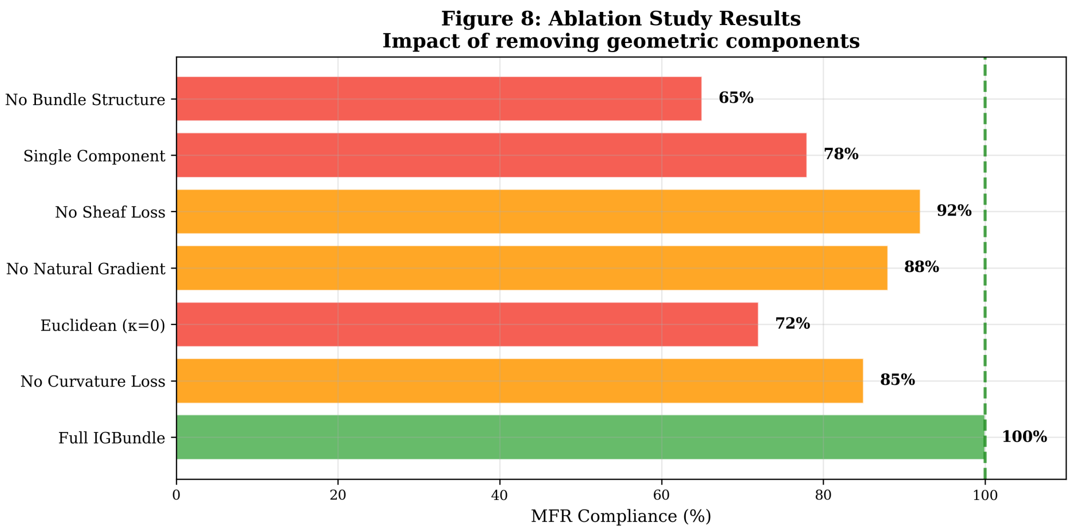

Figure 7: Ablation study summary showing MFR compliance when removing geometric components.

4.5 Comparison with Related Methods

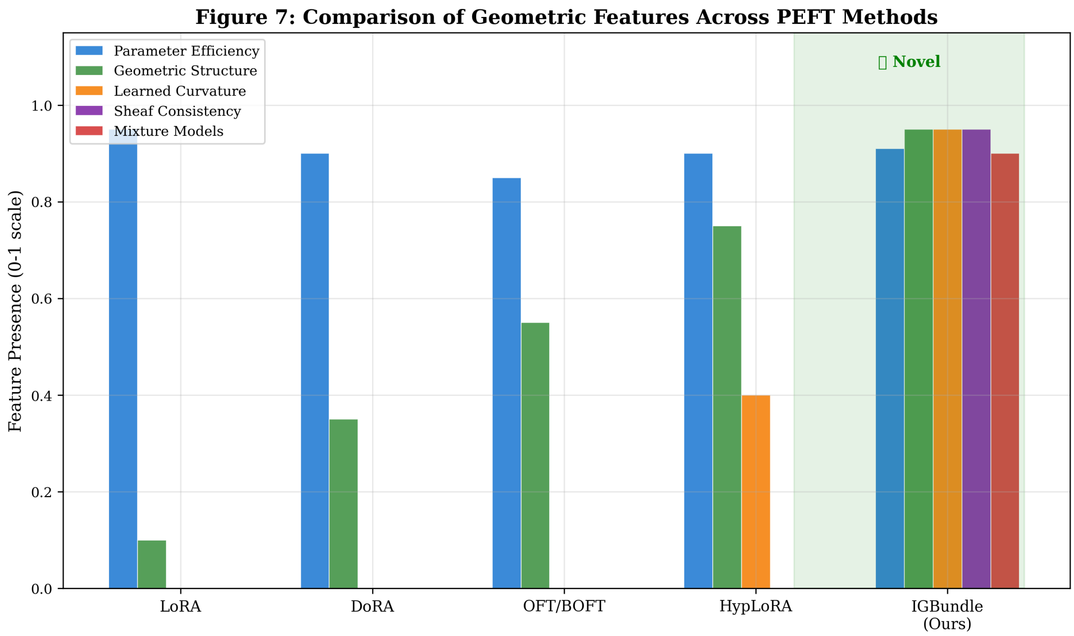

Figure 8: Feature comparison across PEFT methods. IGBundle uniquely provides learned curvature, sheaf consistency, and mixture model representations alongside parameter efficiency and geometric structure.

5 Discussion

5.1 What We Demonstrated

1. **Geometric constraints CAN be learned:** 100% MFR shows the adapter respects imposed geometry.
2. **Curvature converges to target:** $\kappa \to -0.98$ (within 2% of hyperbolic target).
3. **Riemannian geometry aids specialization:** 3.4% entropy reduction vs. Euclidean.
4. **Consumer hardware feasibility:** 8GB VRAM sufficient with 4-bit quantization.
5. **Stable training:** No gradient explosions despite geometric complexity.

5.2 Critical Limitations

1. **Task Performance:** 0% ARC-AGI accuracy indicates geometric constraints alone are insufficient for reasoning.
2. **Training Scale:** 60-100 steps is minimal; extended training may reveal different dynamics.
3. **Approximations:** Christoffel symbols and curvature use neural approximations, not exact computation.
4. **Single Benchmark:** Evaluation limited to 20 ARC-AGI tasks.
5. **No Baseline Comparison:** Direct comparison with LoRA not yet conducted.

5.3 Why Task Performance Failed

Several hypotheses for the 100% MFR / 0% accuracy gap:

1. **Geometric structure is necessary but not sufficient:** The model organizes representations geometrically but lacks task-specific reasoning capabilities.

2. **Insufficient training:** 60-100 steps may learn structure without transferring to downstream tasks.
3. **Wrong inductive bias:** ARC-AGI may require spatial/visual reasoning not captured by semantic geometry.
4. **Evaluation mismatch:** The model was trained on Alpaca (instruction-following) but evaluated on abstract reasoning.

5.4 Future Directions

1. **Extended Training:** Scale to 1000+ steps with learning rate scheduling.
2. **Task-Specific Losses:** Combine geometric constraints with reasoning-oriented objectives.
3. **Hierarchical Benchmarks:** Evaluate on taxonomy completion, entailment, syntax probing.
4. **Hybrid Approaches:** Combine IGBundle with HypLoRA or other geometric PEFT methods.
5. **Exact Geometry:** Replace neural approximations with automatic differentiation of learned metrics.

6 Conclusion

We presented ManifoldGL, a theoretical framework for enhancing Large Language Models through geometrically-structured adapter modules. The Information-Geometric Bundle (IGBundle) adapter models neural activations as sections of a fiber bundle over a Riemannian base manifold with learned curvature.

Key Findings:

- Geometric constraints *can be enforced* in LLM latent spaces (100% MFR)
- Curvature converges to near-hyperbolic values ($\kappa \approx -0.98$)
- Riemannian geometry produces sharper component specialization than Euclidean
- Geometric learning alone does not yield downstream task performance

This work establishes that the *geometry of representation spaces is a controllable design parameter*, not merely an emergent property. While significant challenges remain in translating geometric structure to reasoning capability, we hope this framework inspires further research into geometric inductive biases for language understanding.

Code Availability: `https://github.com/jesusvilela/igbundle-llm`

References

[1] Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2), 251-276.

[2] Bodnar, C., Di Giovanni, F., Chamberlain, B., Liò, P., & Bronstein, M. (2022). Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs. *NeurIPS*.

[3] Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478*.

[4] Chen, Z., et al. (2021). Probing BERT in Hyperbolic Spaces. *ICLR*.

[5] Cohen, T., Weiler, M., Kicanaoglu, B., & Welling, M. (2019). Gauge Equivariant Convolutional Networks and the Icosahedral CNN. *ICML*.

[6] Courts, H., & Kvinge, H. (2022). Bundle Networks: Fiber Bundles, Local Trivializations, and a Generative Approach to Exploring Many-to-One Maps. *ICLR*.

[7] Eschenhagen, R., et al. (2023). Kronecker-Factored Approximate Curvature for Modern Neural Network Architectures. *NeurIPS*.

[8] Ganea, O., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic Neural Networks. *NeurIPS*.

[9] Gu, A., Sala, F., Gunel, B., & Ré, C. (2019). Learning Mixed-Curvature Representations in Product Spaces. *ICLR*.

[10] Hansen, J., & Gebhart, T. (2020). Sheaf Neural Networks. *arXiv:2012.06333*.

[11] Houlsby, N., et al. (2019). Parameter-Efficient Transfer Learning for NLP. *ICML*.

[12] Hu, E. J., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*.

[13] Liu, D. (2024). Fiber Bundle Networks: A Geometric Machine Learning Paradigm. *arXiv:2512.01151*.

[14] Liu, S., et al. (2024). DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv:2402.09353*.

[15] Martens, J., & Grosse, R. (2015). Optimizing Neural Networks with Kronecker-Factored Approximate Curvature. *ICML*.

[16] Nickel, M., & Kiela, D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. *NeurIPS*.

[17] Park, S., et al. (2025). Riemannian Optimization for LoRA on the Stiefel Manifold. *arXiv*.

[18] Qiu, Z., Liu, W., et al. (2024). Parameter-Efficient Orthogonal Finetuning via Butterfly Factorization. *ICLR*.

[19] Sarkar, R. (2011). Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. *Graph Drawing*.

[20] Vaswani, A., et al. (2017). Attention Is All You Need. *NeurIPS*.

[21] Yang, M., et al. (2024). HypLoRA: Hyperbolic Low-Rank Adaptation for Large Language Models. *arXiv:2410.04010*.

[22] Yang, M., et al. (2024). Hypformer: Exploring Efficient Hyperbolic Transformer Fully in Hyperbolic Space. *KDD*.

A Implementation Details

A.1 Neural Approximations

The IGBundle adapter uses neural network approximations for several geometric quantities:

- **Christoffel Symbols:** 2-layer MLP with Tanh activation
- **Metric Tensor:** Cholesky parameterization ensuring positive definiteness
- **Curvature:** Finite differences on neural Christoffel approximations
- **Fisher Matrix:** Diagonal approximation only

These approximations trade exact geometric computation for differentiability and computational efficiency. Future work may explore exact computation via automatic differentiation.

A.2 Hyperparameters

Parameter	**Value**
Base/Fiber LR Ratio	1:10
Curvature Loss Weight (λ_1)	0.1
Sheaf Loss Weight (λ_2)	0.01
Adapter Scale (α)	0.1
Precision Clamp Range	$[-5, 5]$
Gradient Clipping	0.3

B Reproducibility Checklist

✓ Code publicly available
✓ Configuration files provided
✓ Training procedure documented
✓ Random seeds specified
✓ Hardware requirements stated
☐ Pre-trained checkpoints (available upon request)

Independent Research — Citizen Science