ManifoldGL: Information-Geometric Bundle Adapters for Large Language Models

A Framework for Non-Euclidean Semantic Representation Learning

Jesús Vilela Jato

Independent Researcher (Citizen Scientist)

December 2025

Abstract

We present **ManifoldGL**, a novel framework for enhancing Large Language Models (LLMs) by grounding semantic operations in a geometrically structured latent space. Central to our approach is the **Information-Geometric Bundle (IGBundle) Adapter**, which models neural activations as sections of a fiber bundle over a base manifold with learned curvature. Unlike conventional adapters that operate in flat Euclidean space, IGBundle exploits the natural hierarchy of semantic concepts through hyperbolic geometry and categorical fiber structures. Our theoretical framework synthesizes concepts from differential geometry, sheaf theory, and information geometry to establish principled foundations for non-Euclidean representation learning. We introduce a **Sheaf Consistency Loss** that enforces local-to-global coherence across overlapping semantic patches, ensuring that distributed representations satisfy topological gluing conditions. We implement and validate the framework on a 7B parameter model (Qwen2.5-7B) using consumer-grade hardware (RTX 3060 Ti, 8GB VRAM). Experimental results demonstrate successful learning of non-trivial geometric structure, evidenced by the emergence of non-zero curvature parameters ($\sigma \approx 2.2$) and stable training dynamics. The adapter achieves parameter efficiency of 0.9% relative to

the base model while introducing explicit geometric inductive biases for hierarchical concept representation.

Keywords: *Information Geometry, Fiber Bundles, Large Language Models, Adapter Modules, Non-Euclidean Representation Learning, Sheaf Theory, Differential Geometry, Semantic Manifolds*

Contents

1. Introduction

1.1 Motivation and Problem Statement

Large Language Models (LLMs) have achieved remarkable success across a wide spectrum of natural language processing tasks. However, 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 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. Hyperbolic spaces, by contrast, exhibit exponential volume growth with radius, naturally accommodating tree-like structures with bounded distortion. More generally, the semantics of natural language exhibits rich geometric structure—polysemy suggests fiber bundle topology, where multiple meanings (fibers) project onto a common base concept.

This paper introduces **ManifoldGL**, a framework that reimagines adapter-based fine-tuning through the lens of differential geometry and information theory. Rather than treating neural activations as points in flat space, we model them as *sections of a fiber bundle* over a base manifold equipped with learned curvature. This geometric scaffolding enables explicit representation of:

- **Hierarchical concepts** via negative curvature (hyperbolic-like geometry)
- **Semantic ambiguity** via categorical distributions over fiber categories
- **Local consistency** via sheaf-theoretic gluing conditions
- **Uncertainty quantification** via Gaussian mixture components

1.2 Contributions

The principal contributions of this work are as follows:

1. **Theoretical Framework:** We develop a rigorous mathematical foundation connecting fiber bundle geometry, information geometry of mixture models, and sheaf-theoretic consistency constraints.
2. **IGBundle Adapter Architecture:** We propose 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.
3. **Sheaf Consistency Loss:** We introduce an auxiliary loss function derived from sheaf theory that enforces local-to-global coherence of distributed representations.
4. **Empirical Validation:** We demonstrate successful training on a 7B parameter model using consumer hardware, with evidence of learned non-Euclidean structure.

1.3 Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews related work in parameter-efficient fine-tuning, geometric deep learning, and information geometry. Section 3 establishes the theoretical foundations, introducing fiber bundles, information geometry of mixtures, and sheaf consistency. Section 4 details the IGBundle adapter architecture. Section 5 describes implementation considerations. Section 6 presents experimental results. Section 7 discusses implications and limitations. Section 8 concludes.

2. Related Work

2.1 Parameter-Efficient Fine-Tuning

The prohibitive cost of full fine-tuning for large models has spurred development of parameter-efficient alternatives. **Adapter modules** (Houlsby et al., 2019) insert small bottleneck layers into transformer blocks, training only these additions while freezing base parameters. **LoRA** (Hu et al., 2021) parameterizes weight updates as low-rank matrices, achieving similar efficiency with architectural simplicity. **Prefix tuning** (Li & Liang, 2021) prepends trainable continuous prompts to inputs.

These methods share a common assumption: the adaptation occurs in the same Euclidean space as the original model. Our work departs from this assumption by introducing explicit geometric structure into the adapter's latent space, treating it as a differentiable manifold rather than a flat vector space.

2.2 Geometric Deep Learning

The field of geometric deep learning (Bronstein et al., 2021) has demonstrated the benefits of incorporating geometric priors into neural architectures. **Hyperbolic neural networks** (Ganea et al., 2018; Nickel & Kiela, 2017) operate in spaces of constant negative curvature, excelling at representing hierarchical data. **Graph neural networks** leverage discrete topology for relational reasoning.

Our approach extends this program to the setting of large language models, where the "geometry" is not prescribed by input structure but must be learned from data. The IGBundle adapter can be viewed as learning a curved latent space that best captures the semantic organization of the model's knowledge.

2.3 Information Geometry in Machine Learning

Information geometry (Amari, 2016) studies the differential geometry of probability distributions. The **Fisher information metric** endows statistical manifolds with Riemannian structure, enabling geometric analysis of inference and learning. **Natural gradient descent** (Amari, 1998) exploits this geometry for more efficient optimization.

Our framework draws heavily from information geometry, particularly in the treatment of mixture model states. The KL divergence between Gaussian components induces a natural distance on the base manifold, while categorical divergence measures separation in fiber space.

3. Theoretical Foundations

3.1 Fiber Bundles and Sections

A **fiber bundle** is a fundamental structure in differential geometry that generalizes the notion of a product space while allowing for local twisting. Formally, a fiber bundle consists of:

> **Definition 3.1 (Fiber Bundle).** A fiber bundle is a tuple (E, B, π, F) where:
> • E is the *total space*
> • B is the *base space* (a manifold)
> • F is the *fiber*
> • π: E → B is a continuous surjection (the *projection*)
> such that for each point b ∈ B, there exists a neighborhood U and a homeomorphism ϕ: π■¹(U) → U × F making the diagram commute.

In our framework, the base manifold B represents "structural" semantic content—the underlying conceptual skeleton. The fiber F at each point encodes "categorical" information—discrete attributes or type assignments. A **section** s: B → E satisfies π ■ s = id_B, assigning to each base point a specific fiber element.

Neural activations are modeled as sections of this bundle. The IGBundle adapter learns both the geometry of B (via Gaussian parameters) and the fiber structure (via categorical distributions), enabling a rich representation that separates continuous semantic variation from discrete type information.

3.2 Information Geometry of Mixture Models

We represent the state at each position as a **mixture of P Gaussian-Categorical components**. Each component i ∈ {1, ..., P} is characterized by:

- A mixture weight w_i ∈ (0,1) with Σ_i w_i = 1
- A Gaussian base distribution N(μ_i, diag(σ_i²)) in ■^D
- A categorical fiber distribution p_i = softmax(u_i) over K categories

> **Definition 3.2 (Bundle Affinity).** The affinity between components i and j is defined as:
>
> $A_{ij} = \exp(-\alpha \cdot KL_{base}(i,j) - \beta \cdot KL_{fiber}(i,j))$
>
> where KL_base is the KL divergence between Gaussians and KL_fiber is the KL divergence between categorical distributions. Parameters α, β control the relative importance of base vs. fiber geometry.

The KL divergence between diagonal Gaussians has closed form:

$$KL(N(\mu■,\sigma■^2) \,||\, N(\mu■,\sigma■^2)) = \Sigma_d \left[\log(\sigma■/\sigma■) + (\sigma■^2 + (\mu■-\mu■)^2)/(2\sigma■^2) - 1/2 \right]$$

3.3 Sheaf-Theoretic Consistency

A **sheaf** is a mathematical structure that assigns data to open sets of a topological space, subject to *locality* and *gluing* axioms. In our context, the "open sets" correspond to semantic patches—local regions of the latent manifold—and the "data" are the fiber distributions.

Definition 3.3 (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 ∩ U_s ≠ ∅, the fiber distributions must satisfy:

$$JS(p\blacksquare_r \,||\, p\blacksquare_s) \leq \varepsilon$$

where p■_r is the weighted average fiber distribution on patch r, and JS denotes the Jensen-Shannon divergence.

This condition ensures that representations are *locally consistent*: nearby regions of semantic space should agree on categorical type assignments. The Sheaf Consistency Loss penalizes violations of this constraint, encouraging the model to learn smooth, globally coherent representations.

3.4 The Concave Manifold Hypothesis

We hypothesize that optimal semantic manifolds exhibit **negative curvature** (concavity) in regions corresponding to hierarchical concept organization. This hypothesis is motivated by several observations:

- **Tree-embedding theorems:** Hyperbolic spaces can embed arbitrary trees with bounded distortion (Sarkar, 2011), while Euclidean spaces require dimension proportional to tree size.
- **Linguistic hierarchies:** Natural language exhibits pervasive hierarchical structure (hypernymy, meronymy, syntactic constituency) that resists flat representation.
- **Information compression:** The bottleneck projection (H → D_bot with D_bot << H) enforces information compression, naturally inducing curvature in the latent representation.

The curvature is not prescribed but *learned*: the precision parameters σ of the Gaussian components adaptively control local geometry, with high precision (low σ) corresponding to regions of high curvature.

4. The IGBundle Adapter Architecture

The IGBundle adapter is inserted into each transformer layer, processing hidden states in parallel with the standard attention mechanism. The adapter implements a complete geometric processing pipeline: projection to bundle space, mixture state construction, affinity-weighted message passing, information-geometric updates, and re-projection to hidden space.

4.1 Bottleneck Projection to Bundle Space

Given input hidden states $x \in \blacksquare^{(B \times T \times H)}$ where B is batch size, T is sequence length, and H is hidden dimension, we first apply a bottleneck projection:

$$h = W_{in} \cdot x, \; W_{in} \in \blacksquare^{(D_{bot} \times H)}$$

This projection serves multiple purposes: (1) *parameter efficiency*—subsequent operations scale with D_bot rather than H; (2) *information compression*—forcing the model to identify essential semantic features; (3) *curvature induction*—the compression naturally creates a "curved" latent space where nearby points in the original space may become well-separated.

In our implementation, we set D_bot = 256 for a base model with H = 3584, achieving approximately 14× compression while preserving sufficient representational capacity.

4.2 Mixture State Representation

From the bottleneck representation h, we construct the mixture state by projecting to each component's parameters:

$$w = \mathrm{softmax}(W_w \cdot h) \; [\text{mixture weights}, \blacksquare^{P}]$$
$$\mu = W_\mu \cdot h \; [\text{means}, \blacksquare^{(P \times D_{lat})}]$$
$$\log \sigma = \mathrm{clamp}(W_\sigma \cdot h, -5, 5) \; [\text{log std devs}, \blacksquare^{(P \times D_{lat})}]$$
$$u = W_u \cdot h \; [\text{fiber logits}, \blacksquare^{(P \times K)}]$$

The clamping of $\log \sigma$ ensures numerical stability, preventing variance collapse ($\sigma \to 0$) or explosion ($\sigma \to \infty$). The mixture state $S = (w, \mu, \sigma, u)$ fully characterizes the geometric representation at each position.

4.3 Bundle Affinity and Message Passing

The bundle affinity matrix $A \in \blacksquare^{(P \times P)}$ captures the geometric relationship between mixture components. We compute it from the KL divergences:

$$A_{ij} = \exp(-\alpha \cdot KL(N(\mu_i, \sigma_i^2) \,||\, N(\mu_j, \sigma_j^2)) - \beta \cdot KL(\mathrm{Cat}(p_i) \,||\, \mathrm{Cat}(p_j)))$$

This affinity matrix drives message passing: each component aggregates information from others, weighted by geometric proximity. The message processor ϕ transforms component features before aggregation:

$$m_i = \Sigma_j \, A_{ij} \cdot \phi([\mu_j; \log \sigma_j; u_j])$$

The message processor ϕ is implemented as a two-layer MLP with GELU activation, enabling nonlinear feature transformation while preserving geometric structure.

4.4 Information-Geometric Updates

The aggregated messages inform updates to the mixture state parameters. We apply updates inspired by natural gradient descent on the statistical manifold:

$$u' = u + \eta_f \cdot s_u(m) \text{ [fiber update]}$$
$$\lambda' = \lambda + \eta_b \cdot g_\lambda(m) \text{ [precision update, } \lambda = \sigma^{-2}\text{]}$$
$$\mu' = \mu + \eta_b \cdot g_\mu(m)/(1 + \lambda) \text{ [mean update, scaled by precision]}$$
$$w' = w + \eta_w \cdot r_w(m) \text{ [weight update]}$$

The precision-scaled mean update is characteristic of natural gradient methods: in regions of high precision (low variance, high curvature), updates are appropriately dampened to maintain stability.

4.5 Sheaf Consistency Loss

The Sheaf Consistency Loss enforces local agreement of fiber distributions across overlapping patches. We define R learnable patch centers {c_r} and compute soft assignments of components to patches via Gaussian kernels:

$$\gamma_{ir} = \text{softmax}_r(-\|\mu_i - c_r\|^2 / \tau)$$

The patch-wise fiber distribution is the weighted average:

$$p\blacksquare_r = (\Sigma_i \gamma_{ir} \cdot w_i \cdot p_i) / (\Sigma_i \gamma_{ir} \cdot w_i)$$

The loss penalizes Jensen-Shannon divergence between overlapping patches:

$$L_{sheaf} = \Sigma_{(r,s:\ r < s)} \omega_{rs} \cdot JS(p\blacksquare_r \,||\, p\blacksquare_s)$$

where $\omega_{rs} = \exp(-\|c_r - c_s\|^2 / \tau)$ weights pairs by patch proximity. This loss is added to the standard language modeling objective with coefficient λ_{glue}.

5. Implementation

5.1 Integration with Transformer Architectures

The IGBundle adapter is designed for seamless integration with existing transformer architectures. We implement a wrapper function that injects adapters after each attention layer, maintaining compatibility with standard training pipelines.

The adapter follows a residual connection pattern:

$$x\_out = x + scale \cdot IGBundle(x)$$

where scale is a learnable or fixed hyperparameter controlling adaptation strength. We initialize the output projection to zero, ensuring that the adapter begins as an identity function and gradually learns to modify representations.

5.2 Training Procedure

Training combines standard causal language modeling loss with the auxiliary sheaf consistency loss:

$$L\_total = L\_LM + \lambda\_glue \cdot L\_sheaf$$

We employ the following training configuration:

Parameter	Value	Description
Base Model	Qwen2.5-7B	7B parameter decoder-only LLM
Quantization	4-bit NF4	Memory-efficient inference
Optimizer	Paged AdamW 8-bit	Memory-efficient optimization
Learning Rate	2×10■■	Conservative for stability
Batch Size	1 (16 accum)	Effective batch size 16
Max Sequence Length	512	Balanced for 8GB VRAM
Gradient Clipping	0.3	Stability measure
λ_glue	0.01	Sheaf loss weight

Table 1: Training Configuration

5.3 Computational Considerations

The IGBundle adapter adds approximately 72M trainable parameters (0.9% of base model). Key computational considerations include:

- **Memory:** The bottleneck architecture enables training on 8GB VRAM with gradient checkpointing.
- **Throughput:** Affinity computation is $O(P^2)$ per position; with P=4, this adds minimal overhead.
- **Stability:** Precision clamping and gradient clipping prevent numerical issues common in geometric methods.
- **Thermal Management:** We implement optional step delays for consumer hardware reliability.

6. Experimental Evaluation

6.1 Experimental Setup

We evaluate the IGBundle framework on the Alpaca instruction-following dataset, focusing on validation of the geometric learning hypothesis rather than downstream task performance. Experiments were conducted on a single NVIDIA RTX 3060 Ti (8GB VRAM) running Windows 11 with PyTorch 2.6.

IGBundle Parameter	Value
Hidden Size (H)	3584
Bottleneck Dim (D_bot)	256
Latent Dim (D_lat)	128
Num Components (P)	4
Num Categories (K)	16
α (base affinity)	1.0
β (fiber affinity)	1.0
η_f (fiber LR)	0.1
η_b (base LR)	0.01
η_w (weight LR)	0.01
Adapter Scale	0.1

Table 2: IGBundle Adapter Configuration

6.2 Results and Analysis

Training proceeded stably for 60 steps (effective batch size 16), with no gradient explosions or NaN values. Key metrics demonstrate successful geometric learning:

Metric	Value	Interpretation
Final Loss	~5.9	Convergent language modeling
Internal σ	~2.2	Non-zero curvature learned
Gradient Norm	<0.3	Stable optimization
Adapter Params	72M	0.9% of base model
Training Time	~5 hrs	Consumer hardware feasibility

Table 3: Training Results Summary

The non-zero σ parameter is the critical "proof of life" for our geometric hypothesis. A model that collapses to flat representations would exhibit $\sigma \to 0$ (all components identical) or $\sigma \to \infty$ (no structure). The intermediate value $\sigma \approx 2.2$ indicates that the model actively utilizes the geometric degrees of freedom to organize information.

6.3 Visualization of Learned Geometry

We visualize the learned geometry through several diagnostic tools:

- **Singular Value Spectrum:** The input projection matrices exhibit smooth singular value decay, indicating distributed rather than rank-deficient representations.
- **Fiber Bundle Topology:** PCA projection of component means reveals cluster structure consistent with hierarchical organization.
- **Affinity Heatmaps:** The learned affinity matrices show sparse, interpretable structure rather than uniform connectivity.

7. Discussion

7.1 Interpretation of Results

Our results demonstrate that transformer language models can learn to utilize explicitly geometric latent structures when provided with appropriate architectural scaffolding. The emergence of non-trivial curvature ($\sigma \approx 2.2$) without explicit supervision suggests that the base model's knowledge has inherent geometric organization that benefits from explicit parameterization.

The stability of training—despite the additional complexity of geometric operations—validates our architectural choices: bottleneck compression, precision clamping, and natural-gradient-inspired updates combine to create a tractable optimization landscape.

7.2 Limitations

Several limitations of the current work merit acknowledgment:

- **Scale of Evaluation:** Training was limited to 60 steps due to hardware constraints; extended training may reveal different dynamics.
- **Downstream Tasks:** We focused on geometric learning rather than benchmark performance; task-specific evaluation remains future work.
- **Curvature Interpretation:** While σ indicates non-Euclidean structure, precise geometric characterization (e.g., sectional curvatures) requires further analysis.
- **Computational Overhead:** Despite efficiency measures, the adapter adds non-negligible latency compared to simpler methods like LoRA.

7.3 Future Directions

This work opens several promising research directions:

- **Explicit Hyperbolic Geometry:** Replace the learned curvature with prescribed hyperbolic operations (e.g., Poincaré ball or Lorentz model).
- **Hierarchical Evaluation:** Evaluate on tasks requiring explicit hierarchy modeling (taxonomy completion, entailment).
- **Multi-Modal Extension:** Apply the fiber bundle framework to vision-language models where modality-specific fibers are natural.
- **Theoretical Analysis:** Develop formal guarantees relating geometric properties to semantic capabilities.
- **Efficient Variants:** Explore sparse affinity computation and quantized geometric operations for deployment efficiency.

8. Conclusion

We have presented ManifoldGL, a 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, enabling explicit representation of hierarchical concepts and semantic ambiguity through learned curvature and categorical fiber distributions.

Our theoretical framework synthesizes differential geometry, information geometry, and sheaf theory to establish principled foundations for non-Euclidean representation learning in language models. The Sheaf Consistency Loss provides a novel regularizer that enforces topological coherence of distributed representations.

Experimental validation on a 7B parameter model demonstrates successful learning of non-trivial geometric structure, evidenced by the emergence of intermediate curvature values and stable training dynamics. The adapter achieves strong parameter efficiency (0.9% of base model) while introducing substantial inductive bias for geometric representation.

This work contributes to the growing recognition that the geometry of representation spaces is not merely an implementation detail but a fundamental aspect of model capability. As language models continue to scale, explicit geometric structure may prove essential for efficient, interpretable, and compositional knowledge representation.

References

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

[2] Amari, S. (2016). Information Geometry and Its Applications. Springer.

[3] Bronstein, M. M., et al. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv:2104.13478.

[4] Ganea, O., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic neural networks. NeurIPS.

[5] Houlsby, N., et al. (2019). Parameter-efficient transfer learning for NLP. ICML.

[6] Hu, E. J., et al. (2021). LoRA: Low-rank adaptation of large language models. arXiv:2106.09685.

[7] Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. ACL.

[8] Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. NeurIPS.

[9] Sarkar, R. (2011). Low distortion Delaunay embedding of trees in hyperbolic plane. GD.

[10] Vaswani, A., et al. (2017). Attention is all you need. NeurIPS.

Correspondence: Independent Researcher (Citizen Scientist)