GraphShaper: Geometry-aware Alignment for Improving Transfer Learning in Text-Attributed Graphs

Heng Zhang
South China Normal University
China
2024025450@m.scnu.edu.cn

Tianyi Zhang
Uber Technologies Inc.
USA
tianyizhg@gmail.com

Yuling Shi
Shanghai Jiao Tong University
China
yuling.shi@sjtu.edu.cn

Xiaodong Gu
Shanghai Jiao Tong University
China
xiaodong.gu@sjtu.edu.cn

Yaomin Shen
Nanchang Research Institute,
Zhejiang University
China
coolshennf@gmail.com

Haochen You
Columbia University
USA
hy2854@columbia.edu

Zijian Zhang
University of Pennsylvania
USA
zzjharry@alumni.upenn.edu

Yilei Yuan
University of Michigan
USA
yiliey@umich.edu

Jin Huang*
South China Normal University
China
huangjin@m.scnu.edu.cn

Abstract

Graph foundation models represent a transformative paradigm for learning transferable representations across diverse graph domains. Recent methods leverage large language models to unify graph and text modalities into a shared representation space using contrastive learning. However, systematic evaluations reveal significant performance degradation at structural boundaries where distinct topological patterns converge, with accuracy losses exceeding 20 percentage points. This issue arises from a key limitation: current methods assume all graph structures can be encoded within a single Euclidean space. In reality, tree structures require hyperbolic geometry to preserve hierarchical branching, while cyclic patterns depend on spherical geometry for closure properties. At structural boundaries, nodes experience conflicting geometric constraints that uniform encoding spaces cannot resolve. This raises a crucial challenge: **Can alignment frameworks be designed to respect the intrinsic geometric diversity of graph structures?** We introduce **GraphShaper**, a geometry-aware framework that enhances graph encoding through multi-geometric specialization. Our approach employs expert networks tailored to different geometric spaces, dynamically computing fusion weights to adaptively integrate geometric properties based on local structural characteristics. This adaptive fusion preserves structural integrity before alignment with text embeddings. Extensive experiments demonstrate that GraphShaper achieves 9.47% accuracy improvements on citation networks and 7.63% on social networks in zero-shot settings.

CCS Concepts

• **Information systems** → **Data mining**.

Keywords

Graph Foundation Model, Graph Transformer, Graph Representation Learning, Self-supervised Learning

1 Introduction

Text-Attributed Graphs (TAGs) have become an important representation for capturing complex systems by combining structural connectivity with rich textual features [55]. Recent advances in integrating graph learning with large language models offer new approaches for reasoning over these hybrid structures [21]. Self-supervised pre-training methods have emerged as key techniques, enabling models to learn transferable representations from large-scale graph data [59]. Contrastive learning approaches unify graph and text modalities into shared representation spaces, achieving effective alignment between structural and textual signals [1]. These methods have demonstrated strong zero-shot and few-shot performance, showcasing the potential of TAGs in cross-domain knowledge transfer and task generalization [17].

Approaches to integrating language models with graph learning can be divided into three strategies [11]. The first focuses on enhancing node attributes by using language models to extract rich textual features, refining node representations beyond shallow embedding methods [27, 52]. Pre-trained encoders and domain-specific prompting are often used to unify diverse downstream tasks [3, 8]. The second treats language models as predictors for graph tasks by converting graph structures into token sequences, enabling models to process graph-derived tokens in a text-comprehensible format [31]. The third emphasizes aligning graph and text modalities via contrastive learning, optimizing shared embeddings that

*Corresponding author.

Conference'17, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

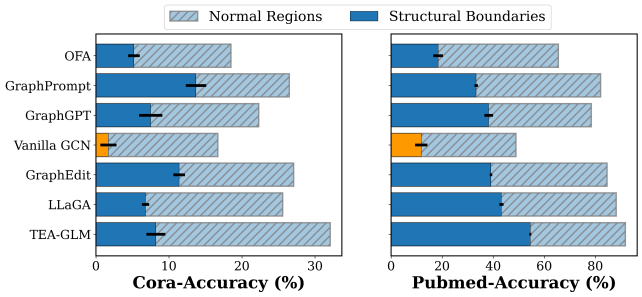


Figure 1: Performance degradation at structural boundaries. We compare zero-shot accuracy on normal regions versus structural boundaries across Cora and Pubmed. All methods exhibit systematic performance drops at boundaries.

unify multimodal information [21]. These strategies have achieved notable results in cross-domain transfer and zero-shot tasks [5, 61], yet they process diverse graph structures through uniform encoding mechanisms, typically constrained to Euclidean geometric operations [60]. This raises questions about their ability to capture the geometric diversity inherent in real-world graphs, motivating systematic empirical investigation [13, 15].

To assess the robustness of current graph foundation models, we conducted analyses across structurally diverse regions from citation, social, and e-commerce networks. Our findings revealed an intriguing pattern: model performance varied substantially across different graph regions. This variation was not random but closely linked to local structural properties. In regions with uniform structural patterns, models performed consistently well. However, accuracy dropped sharply, often exceeding 20 percentage points, at transition zones where structural patterns changed. These degradation zones consistently appeared at the boundaries between differing topological motifs. Across all tested graphs, similar performance declines were observed in regions where distinct structural patterns converged. Such variations could not be explained by simple metrics such as density or degree alone. Nodes near structural boundaries were statistically similar to those in homogeneous regions when measured by local statistics, suggesting that the degradation arose from deeper structural mechanisms. The consistent patterns of performance drops across various domains and model architectures indicate that this is not merely a limitation of model capacity. Instead, it highlights the challenges uniform encoding mechanisms face when handling structural heterogeneity. Understanding why structural boundaries differ from homogeneous regions is key to addressing this issue.

Recent developments in geometric graph learning have shown that different structural patterns align naturally with specific geometric spaces. Tree structures rely on hyperbolic geometry to preserve exponential volume growth without distortion. Cyclic patterns require spherical geometry to maintain rotational symmetry and closure. Regular grids are well-suited to Euclidean geometry for uniform local connectivity. These geometric properties impose constraints on how structural information is preserved during embedding. However, current graph-text alignment methods compress all patterns into a single Euclidean framework, creating conflicts at boundary regions. Suppose a node connects both to a tree-like structure and a cyclic motif. Preserving the tree's hierarchy demands exponential decay in distances to descendant nodes, while maintaining the cyclic closure requires bounded cumulative distances along the cycle. These requirements are inherently incompatible in a single Euclidean space. The encoder must prioritize one property at the expense of the other, leading to inevitable distortions. This trade-off during encoding explains the performance drops shown in Figure 1. The distortions introduced when embedding also impact transferability. Models pre-trained on graphs with specific structural compositions learn a trade-off strategy tailored to that pattern distribution. When applied to graphs with different compositions, the nature of boundary conflicts changes. Consequently, the pre-trained strategy becomes suboptimal. This explains why existing models perform well within their training domain but struggle during cross-domain transfer.

Inspired by the preceding observations, we introduce GraphShaper, a geometry-aware framework designed to resolve the geometric conflicts inherent in current graph-text alignment methods. Unlike existing approaches that apply uniform encoding mechanisms to all structural patterns, GraphShaper improves graph representations by incorporating specialized modules tailored to distinct geometric properties. Each token is processed through networks optimized for Euclidean, hyperbolic, and spherical transformations. The model dynamically computes fusion weights for these modules based on the node's local structural features, allowing it to flexibly integrate multiple geometric attributes. By resolving conflicts through adaptive geometric composition rather than imposing trade-offs, GraphShaper captures the geometric complexity of heterogeneous regions during encoding. This geometry-aware encoding strategically aligns graph representations with text embeddings. To ensure the modules specialize effectively, we encourage diversity by maximizing weight differentiation in hypersphere space. This design enables GraphShaper to develop distinct capabilities across its geometric experts, enhancing its ability to tackle structural heterogeneity. Our contributions can be concluded as:

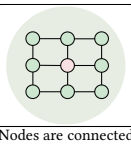
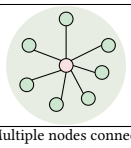
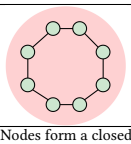
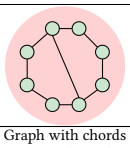
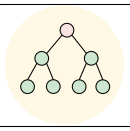
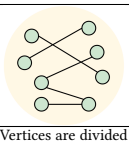
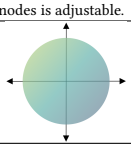
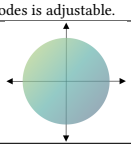
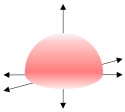
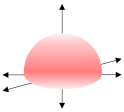
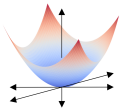
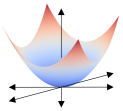
- We reveal that existing TAG methods face geometric limitations by encoding all structures in a single Euclidean space. This restriction prevents them from capturing diverse structural patterns and causes consistent performance drops at structural boundaries.
- We introduce **GraphShaper**, a geometry-aware framework using multi-geometric expert networks to transcend single-space limitations. Through adaptive geometric fusion, it enables robust cross-domain transfer without requiring labeled target data.
- Comprehensive experiments validate GraphShaper's effectiveness, with accuracy gains of **9.47%** on citation networks and **7.63%** on social networks in zero-shot transfer scenarios.

2 Related Work

2.1 Text-Attributed Graph Methods with LLMs

The integration of LLMs into graph learning has explored three primary integration strategies [12, 18, 22]. Enhancement-based methods leverage LLMs to enrich node features with semantic information before feeding them to downstream models [7, 25, 41, 43].

Table 1: Illustrative examples of structural patterns and their geometric correspondences. Different topological patterns exhibit distinct geometric properties. Euclidean geometry suits regular connectivity. Spherical geometry captures cyclic patterns. Hyperbolic geometry accommodates hierarchical structures. Real-world graphs combine multiple patterns.

Structure Type	Grid	Star	Circle	Chordal Cycle	Tree	Bipartite Graph
Illustration						
Description	Nodes are connected in a grid pattern, and the number of nodes is adjustable.	Multiple nodes connect to a central node, and the number of nodes is adjustable.	Nodes form a closed loop, and the number of nodes is adjustable.	Graph with chords in the cycle, with adjustable nodes and chords.	Nodes form a tree, branching and height are adjustable.	Vertices are divided into two disjoint sets, adjustable nodes.
Geometry						
Geometry	Euclidean	Euclidean	Spherical	Spherical	Hyperbolic	Hyperbolic

TAPE [16] uses ChatGPT to enhance node attributes beyond traditional Bag of Words representations. OFA [26] and ZeroG [23] employ language models to unify node features and introduce graph prompting techniques for task standardization. Direct prediction approaches serialize graph structures into natural language, enabling LLMs to perform end-to-end inference [27, 29, 46, 48, 56]. GraphText [58] transforms graphs into text sequences using G-Syntax Trees. GraphGPT [36] and LLaGA [6] encode graph data into tokens through GNNs, though they require labeled data for training and show limited transferability [9, 24]. Alignment-based methods map graph and text modalities into a shared embedding space [5, 54]. GLEM [57] optimizes GNNs and LLMs through iterative training. ConGrat [2] and G2P2 [44] focus on node-text contrastive pretraining but lack graph summary text information. GraphCLIP [61] bridges GNN representations with LLM embeddings, achieving impressive zero-shot performance across diverse tasks. Beyond these primary paradigms, LLMs serve various auxiliary roles [42, 53]. GraphEdit [14] employs them for data augmentation, RLMRec [33] aligns GNN embeddings with linguistic knowledge, and ENG [62] generates synthetic nodes to address data scarcity.

2.2 Low-resource Graph Learning

Real-world graph applications often face limited labeled data, making few-shot and zero-shot learning crucial for practical deployment [32, 36, 51]. Early approaches leverage meta-learning paradigms to enable rapid adaptation with minimal examples [63, 64]. Self-supervised methods including DGI [37] and GraphCL [50] construct contrastive views to enhance node representations through pretraining, while MVGRL [15] incorporates subgraph and diffusion information to capture richer graph semantics. However, these methods typically require task-specific fine-tuning and degrade significantly when supervision becomes extremely sparse. Recent advances explore pseudo-labeling techniques where expand limited labeled data through confident predictions on unlabeled nodes [10, 17, 49]. The emergence of LLMs offers new possibilities for addressing these limitations [40]. LLaGA [6] introduces encoding strategies that enable LLMs to process graph data without training, while TEA-GLM [38] aligns GNN representations with LLM token embeddings to achieve state-of-the-art cross-task and cross-dataset generalization. These developments demonstrate that LLMs can effectively tackle graph learning challenges without task-specific supervision [4, 28, 61].

3 Preliminaries

3.1 Text-Attributed Graphs

A text-attributed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{T})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the node set with $|\mathcal{V}| = N$ nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set capturing structural relationships, $\mathcal{X} \in \mathbb{R}^{N \times d}$ represents node feature matrix derived from textual content, and $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$ denotes the collection of raw text descriptions. The adjacency matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$ encodes the graph topology, where $\mathbf{A}_{ij} = 1$ indicates an edge between nodes v_i and v_j. For large-scale graphs, we employ random walk with restart to extract ego-networks $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{X}_i, \mathcal{T}_i)$ centered around target nodes, capturing local structural and semantic patterns while maintaining computational efficiency.

3.2 Riemannian Geometry

A Riemannian manifold $\mathbb{M}^d$ is a smooth manifold equipped with a Riemannian metric g that defines geometric properties at each point. The curvature of the manifold determines how distances and geodesics behave. Zero curvature corresponds to Euclidean space $\mathbb{E}$ where geodesics remain parallel. Positive curvature defines spherical space $\mathbb{S}$ where geodesics converge. Negative curvature characterizes hyperbolic space $\mathbb{H}$ where geodesics diverge. These distinct geometric spaces provide natural frameworks for representing different structural patterns. Hyperbolic spaces naturally accommodate hierarchical tree structures due to exponential volume growth. Spherical spaces capture cyclic patterns through rotational symmetry and closure properties. Euclidean spaces handle regular grid-like connectivity. The exponential map $\exp_x : T_x\mathbb{M} \to \mathbb{M}$ projects vectors from the tangent space $T_x\mathbb{M}$ onto the manifold.

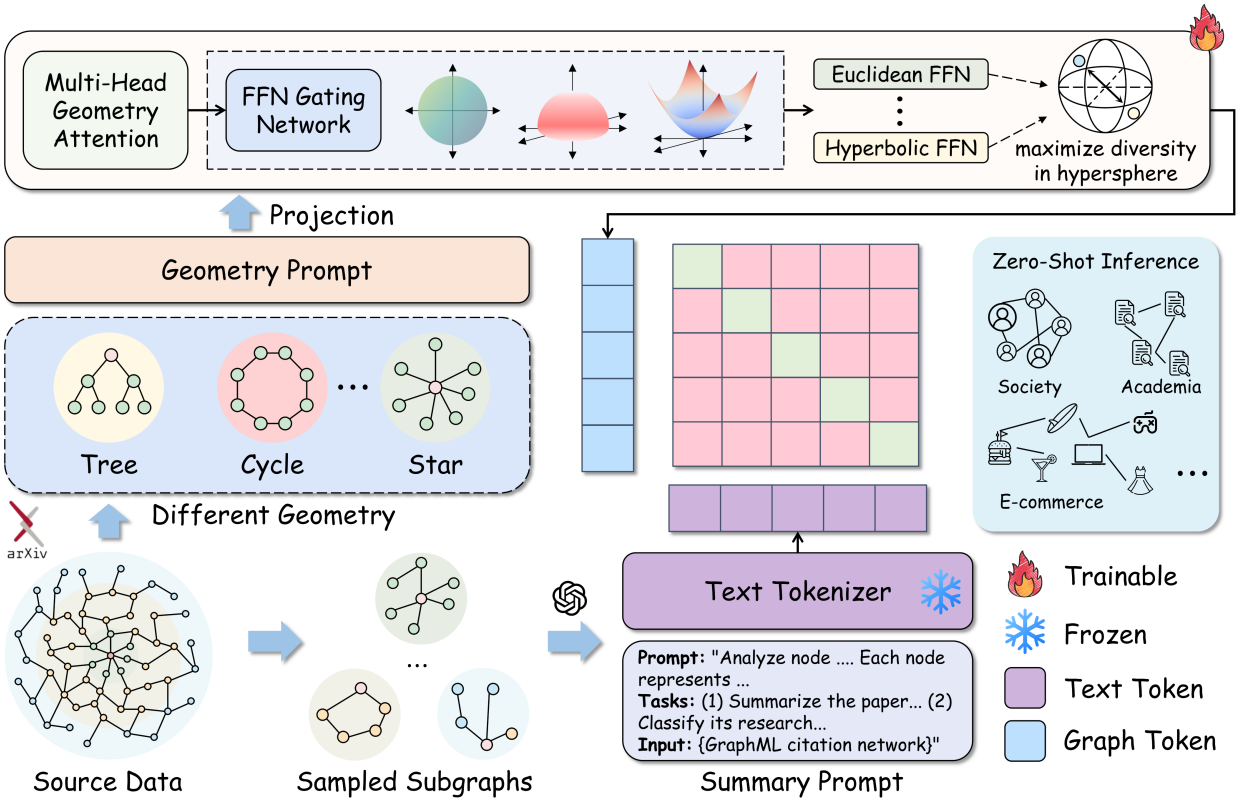


Figure 2: Overview of the GraphShaper framework. The framework captures local structural patterns from input graphs and encodes them across three distinct geometric spaces. Multi-head geometry attention combines these representations via a gating network. Specialized feed-forward experts perform adaptive geometric fusion at structural boundaries. This design enables zero-shot inference across different domains.

The logarithmic map $\log_x : \mathbb{M} \to T_x\mathbb{M}$ performs the inverse operation. These maps enable computations in the tangent space while maintaining geometric properties of the manifold.

4 Methodology

GraphShaper is a geometry-aware alignment framework that addresses geometric representation conflicts in text-attributed graphs through multi-geometric specialization. The framework consists of three key components: a Graph-Summary Pair Generation module that creates semantically rich training data, a Geometry-aware Graph Encoding mechanism that captures local structural patterns in their optimal geometric spaces, and a Geometry-adaptive Alignment component that resolves conflicts through learned geometric composition. To enable both zero-shot and few-shot transfer learning, we augment the architecture with dedicated adaptation mechanisms.

4.1 Graph-Summary Pair Generation

Following established practices in graph-text alignment, we leverage large language models to generate summaries for graph structures, constructing high-quality graph-summary pair training data. We employ GraphML markup language and design prompt templates to enhance comprehension of input graphs. For each sampled subgraph $\mathcal{G}_i$ extracted via random walk with restart sampling, we transform structural and textual information into GraphML format, where nodes contain title and abstract attributes while edges specify relationship types. Using GPT-4 as our summary generator, we create comprehensive graph summaries S_i that capture both local structural patterns and semantic context. This process generates large-scale graph-summary pair data containing over 0.14B tokens across diverse domains including citation networks, e-commerce platforms, and social networks. The resulting paired data $\{(\mathcal{G}_i, S_i)\}_{i=1}^{N}$ provides the foundation for our geometry-aware contrastive pretraining, where N represents the total number of graph-summary pairs and each S_i contains rich semantic descriptions of the corresponding subgraph.

4.2 Geometry-aware Graph Encoding

To capture the geometric diversity inherent in local structural contexts, we develop a geometry-aware encoding mechanism that processes each node's neighborhood in multiple geometric spaces.

For a subgraph $\mathcal{G}_i$ with node features $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$, we first extract the local structural context around each node through k-hop neighborhood sampling.

Multi-geometric Structure Encoding. For each node v, we construct three complementary structural representations based on different geometric spaces. In Euclidean space, we compute local connectivity features capturing grid-like patterns through standard graph convolution. In hyperbolic space, we measure hierarchical relationships and branching structures that characterize tree-like patterns. In spherical space, we compute angular relationships and closure properties for cyclic patterns. As illustrated in figure following, we extract diverse local substructures from the input graph, including trees, cycles, and star patterns, then encode each in its corresponding geometric space.These geometry-specific features are learned through specialized Riemannian graph convolutions. We first map node features to each geometric manifold through coordinate transformation:

$$\mathbf{V}_{\mathbb{M}} = \exp_o^c(\text{proj}(\mathbf{V}_{\text{local}})), \tag{1}$$

where $\mathbb{M} \in \{\mathbb{E}, \mathbb{H}, \mathbb{S}\}$ represents the geometric space, c denotes the curvature parameter, $\text{proj}(\cdot)$ projects to the tangent space, and $\mathbf{V}_{\text{local}}$ represents local structural coordinates derived from the k-hop neighborhood. We then identify structurally similar nodes using K-nearest neighbors based on these geometric coordinates, forming manifold-specific adjacency matrices $\mathbf{A}_{\mathbb{M}}$.

The geometry-aware features $\mathbf{P}_{\mathbb{M}}$ are learned through a two-stage Riemannian GraphSAGE architecture:

$$\begin{cases} \mathbf{Z}_{\mathbb{M}} = \mathcal{F}_{\mathbb{E}\to\mathbb{M}}(\mathbf{X}_i, \mathbf{A}_{\mathbb{M}}), & \text{Step 1} \\ \mathbf{P}_{\mathbb{M}} = \mathcal{F}_{\mathbb{M}\to\mathbb{M}}(\mathbf{Z}_{\mathbb{M}}, \mathbf{A}_{\mathbb{M}}), & \text{Step 2} \end{cases} \tag{2}$$

where $\mathcal{F}_{\mathbb{M}_{\text{in}}\to\mathbb{M}_{\text{out}}}$ represents a Riemannian GraphSAGE layer:

$$\mathcal{F}_{\mathbb{M}_{\text{in}}\to\mathbb{M}_{\text{out}}}(\mathbf{X}, \mathbf{A}) = \exp_o^{c_{\text{out}}}(\text{SAGE}(\log_o^{c_{\text{in}}}(\mathbf{X}), \mathbf{A})), \tag{3}$$

where $\log_o^c$ transforms features from manifold $\mathbb{M}_{\text{in}}$ to the tangent space at origin, SAGE performs GraphSAGE aggregation, and $\exp_o^{c_{\text{out}}}$ maps aggregated features to target manifold $\mathbb{M}_{\text{out}}$. This two-step process first translates Euclidean node features to each target geometric space, then refines representations within that geometric context. The resulting geometry-aware features $\{\mathbf{P}_{\mathbb{E}}, \mathbf{P}_{\mathbb{H}}, \mathbf{P}_{\mathbb{S}}\}$ encode local structural properties from multiple geometric perspectives, enabling each structural pattern to achieve optimal representation quality in its natural geometric space.

4.3 Geometry-adaptive Alignment

As established in our motivation, nodes at structural boundaries face incompatible geometric constraints that cannot be resolved in uniform Euclidean space. To address this, we design a geometry-adaptive alignment mechanism that allows each token to flexibly integrate multiple geometric properties based on its local structural context. Unlike prior work that processes different structures in isolated geometric spaces, our approach enables adaptive geometric fusion through a mixture-of-experts architecture.

The framework integrates three expert FFN layers, each optimized for processing features within its designated geometry while maintaining the conventional Euclidean FFN from pre-trained models:

$$\mathcal{F}_E(\mathbf{x}) = \mathbf{W}_2^{(E)}(\sigma(\mathbf{W}_1^{(E)}\mathbf{x} + \mathbf{b}_1^{(E)})) + \mathbf{b}_2^{(E)}, \tag{4}$$

System prompt: You maintain three geometric representa tions for graph structures. {Euclidean Space} encodes reg ular connectivity patterns with standard distance metrics. {Hyperbolic Space} captures hierarchical tree structures t hrough exponential volume growth. {Spherical Space} m odels cyclic patterns via rotational symmetry. For each no de in the input graph, extract its k-hop neighborhood to co nstruct local structural context. Encode this context in all t hree geometric spaces using specialized graph convolutio ns. Compute geometry-specific features: grid connectivity in Euclidean, branching depth in Hyperbolic, and closure properties in Spherical. Your output should contain three f eature matrices. These multi-geometric representations en able adaptive fusion at structural boundaries.
User content: {GraphML citation network}.

where $\mathbf{W}_1^{(E)}, \mathbf{W}_2^{(E)} \in \mathbb{R}^{d\times d}$ are learnable weight matrices for the Euclidean expert, $\mathbf{b}_1^{(E)}, \mathbf{b}_2^{(E)} \in \mathbb{R}^d$ are bias vectors, and $\sigma(\cdot)$ represents the activation function.

The spherical expert layer captures angular relationships and global symmetries:

$$\mathcal{F}_S(\mathbf{x}) = \kappa \cdot \text{normalize}(\mathbf{W}_2^{(S)}(\sigma(\mathbf{W}_1^{(S)}\mathbf{x} + \mathbf{b}_1^{(S)})) + \mathbf{b}_2^{(S)}), \tag{5}$$

where $\kappa > 0$ controls the spherical curvature parameter, $\text{normalize}(\cdot)$ ensures outputs lie on the unit sphere, and $\mathbf{W}_1^{(S)}, \mathbf{W}_2^{(S)}$ are spherical expert weight matrices with corresponding bias terms $\mathbf{b}_1^{(S)}, \mathbf{b}_2^{(S)}$.

The hyperbolic expert layer processes hierarchical structures:

$$\mathcal{F}_H(\mathbf{x}) = \exp_0(\mathbf{W}_2^{(H)}(\sigma(\log_0(\mathbf{W}_1^{(H)} \otimes \mathbf{x} + \mathbf{b}_1^{(H)}))) + \mathbf{b}_2^{(H)}), \tag{6}$$

where $\exp_0(\cdot)$ and $\log_0(\cdot)$ represent exponential and logarithmic maps at the origin of the Poincaré ball, $\otimes$ denotes Möbius matrix multiplication, and $\mathbf{W}_1^{(H)}, \mathbf{W}_2^{(H)}$ are hyperbolic expert weight matrices.

To enhance expert diversity and prevent parameter redundancy, we maximize weight diversity in hypersphere space. We enforce orthogonality constraints among expert parameters by minimizing their pairwise correlations:

$$\max_{\{\mathbf{W}_1^{(i)}, \mathbf{W}_2^{(i)}\}_{i\in\{E,S,H\}}} \mathcal{L}_{MHS}(\mathbf{W}) = \min_{i\neq j} \rho(\tilde{\mathbf{W}}_i, \tilde{\mathbf{W}}_j), \tag{7}$$

where $\mathcal{L}_{MHS}(\cdot)$ represents the minimum hyperspherical separation loss, $\rho(\cdot,\cdot)$ computes correlation coefficient between weight vectors, $\tilde{\mathbf{W}}_i = \frac{\text{vec}(\mathbf{W}_i)}{\|\text{vec}(\mathbf{W}_i)\|_2}$ denotes the normalized vectorized weight matrix for expert i, and $\text{vec}(\cdot)$ converts matrix parameters into unit hypersphere representations. This constraint ensures each expert develops distinct geometric specializations while maximizing separation distance among expert weight vectors.

Rather than hard-routing to a single expert, each token's representation is computed as a weighted combination of all geometric experts, allowing flexible integration of multiple geometric properties. A learnable gating network $G(\cdot)$ dynamically computes adaptive fusion weights based on geometric characteristics of input features:

$$\mathbf{y} = \sum_{i\in\{E,S,H\}} G_i(\mathbf{x}) \cdot \mathcal{F}_i(\mathbf{x}), \tag{8}$$

where the gating weights are computed via softmax normalization:

$$G_i(\mathbf{x}) = \frac{\exp(\mathbf{w}_i^T \mathbf{x} + b_i)}{\sum_{j \in \{E,S,H\}} \exp(\mathbf{w}_j^T \mathbf{x} + b_j)}, \tag{9}$$

with learnable parameters $\mathbf{w}_i \in \mathbb{R}^d$ and bias b_i for expert i, ensuring $\sum_i G_i(\mathbf{x}) = 1$. The geometry adapter is integrated every $k = 4$ layers in the transformer architecture to maintain computational efficiency while providing sufficient geometric modeling capacity. This adaptive combination mechanism resolves geometric conflicts through learned geometric composition rather than forced trade-offs, enabling nodes at structural boundaries to simultaneously preserve multiple geometric properties.

For the contrastive alignment objective, we employ invariant alignment loss to enhance cross-domain transferability:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(\mathcal{G},S)\sim\mathbb{P}} \sup_{\tau,\tau'\sim\pi} \|\mathbf{h}_{\mathcal{G}} - \mathbf{u}_S\|^2, \tag{10}$$

where $\mathbf{h}_{\mathcal{G}} \in \mathbb{R}^d$ and $\mathbf{u}_S \in \mathbb{R}^d$ represent aligned graph and summary embeddings respectively, π denotes the augmentation distribution, τ and τ' are specific augmentation instances, and the supremum operator ensures robustness against challenging geometric transformations while preserving semantic alignment quality across diverse structural patterns.

4.4 Model Adaptation

For zero-shot learning scenarios, our pretrained model can be directly deployed on target datasets without additional training. We craft prompts incorporating target label information, such as "This paper belongs to {class}" for citation networks. Zero-shot inference is performed by computing cosine similarity between graph embeddings and label embeddings:

$$\hat{y}_i = \arg\max_k \mathbb{E}_{u_k} \text{sim}(\mathbf{h}_i, \mathbf{u}_k), \tag{11}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and $\mathbf{u}_k$ represents the embedding of label k. For few-shot learning, we introduce geometry-aware prompt tuning that maintains frozen pretrained parameters while learning minimal prompt features. The prompt tuning objective aligns with our pretraining approach by incorporating learnable perturbations:

$$\min_{\boldsymbol{\sigma}} \mathbb{E}_{(\mathcal{G},Z)\sim\mathbb{P}^{\text{tar}}} \left[\mathcal{L}_{\text{SCL}} \left(\mathbf{h}_{\mathcal{G}} + \boldsymbol{\sigma}, \mathbf{u}_Z \right) \right], \tag{12}$$

where $\boldsymbol{\sigma}$ represents learnable prompt features, Z denotes label-related sentences, $\mathbb{P}^{\text{tar}}$ signifies target domain distribution, and $\mathcal{L}_{\text{SCL}}$ is supervised contrastive loss treating same-label pairs as positive and different-label pairs as negative. This approach enables effective knowledge transfer with minimal parameter updates while preserving geometric specializations learned during pretraining.

5 Experiments

5.1 Datasets

We conduct comprehensive experiments on seven text-attributed graph datasets spanning academic and e-commerce domains to evaluate model performance under different shot settings. The datasets include Cora, Pubmed, and ArXiv from the academic domain, along with Children, History, Photo, and Products from the e-commerce domain (as show in Table 2). These datasets capture diverse structural patterns and semantic contexts while covering a broad spectrum of graph sizes and structural complexities. We evaluate the models under 0-shot, 1-shot, 3-shot, and 5-shot settings to assess their generalization capabilities and learning efficiency with limited or no labeled data. For node description tasks, we employ Sentence-BERT (SBERT) Score to measure semantic similarity between generated and ground-truth descriptions, where higher scores indicate better alignment with reference text.

Table 2: Statistics of Text-Attributed Graph datasets.

	#Nodes	#Edges	Domain	#C
Cora [47]	2,708	5,429	Academic	7
Pubmed [34]	19,717	44,338	Academic	3
Children [30]	87,229	721,081	E-commerce	10
History [30]	41,551	358,574	E-commerce	12
Photo [30]	48,362	500,928	E-commerce	12
ArXiv [19]	169,343	1,166,243	Academic	40
Products [19]	2,449,029	61,859,140	E-commerce	47

5.2 Baselines

We compare GraphShaper against diverse baseline methods to validate its effectiveness across different paradigms. For graph foundation models, we include state-of-the-art approaches such as OFA [26] for unified graph task modeling, LLaGA [6] for graph-language integration, GraphGPT [36] for graph instruction tuning, GraphEdit [14] for graph structure learning, TEA-GLM [39] for aligning GNN representations with LLM embeddings, RiemannGFM [35] for Riemannian geometry-based structural vocabulary learning, and GraphCLIP [61] for enhancing transferability in graph foundation models. These methods represent the forefront of graph-text alignment research. We also incorporate traditional graph neural networks including GCN [20] and NodeFormer [45] as baselines to highlight the importance of integrating language models and geometry-aware design in addressing cross-domain challenges.

5.3 Implementation Details

GraphShaper is implemented with a geometry-aware alignment framework tailored for cross-domain transfer learning, with a graph encoder consisting of 12-layer Graph Transformer modules (hidden dimension 1024) and a 6-layer SBERT text encoder (hidden dimension 384). Geometric diversity is achieved using curvature parameters set to $c_{\mathbb{H}} = -1.0$ for hyperbolic space and $c_{\mathbb{S}} = 1.0$ for spherical space, alongside geometry adapters inserted after every four transformer layers. Training is performed over 30 epochs on source datasets using the AdamW optimizer with a learning rate of 1×10^{-5}, a weight decay of 1×10^{-5}, and a batch size of 800 per GPU across 8 A100 80G SXM4 GPUs. Few-shot experiments freeze pretrained parameters while fine-tuning prompt features for 100 epochs with a learning rate of 1×10^{-4}. To ensure performance and efficiency, mixed precision (fp16) is utilized for training, while modal inputs are processed in full precision (fp32) for stability. Furthermore, embeddings are normalized through L2 regularization, with learnable scaling factors α_m incorporated to balance modal contributions.

Table 3: Node classification accuracy ($\% \pm \sigma$) under different shot setting. Bolded results indicate the best performance.

	Dataset	*GNN as predictor*			*LLM as predictor*				
	K-shot	GCN	NodeFormer	OFA	GraphGPT	GraphEdit	LLaGA	TEA-GLM	**GraphShaper**
Cora	0-shot	$1.70_{\pm 0.41}$	$2.40_{\pm 0.94}$	$13.00_{\pm 0.44}$	$12.50_{\pm 0.28}$	$18.40_{\pm 0.41}$	$15.80_{\pm 0.57}$	$19.20_{\pm 0.19}$	$\mathbf{29.80}_{\pm 0.72}$
	1-shot	$7.87_{\pm 0.80}$	$9.32_{\pm 0.35}$	$21.20_{\pm 0.47}$	$16.48_{\pm 0.24}$	$23.67_{\pm 0.79}$	$24.62_{\pm 0.34}$	$30.35_{\pm 1.02}$	$\mathbf{33.45}_{\pm 0.91}$
	3-shot	$13.31_{\pm 0.66}$	$14.51_{\pm 0.99}$	$17.12_{\pm 0.06}$	$20.55_{\pm 0.71}$	$26.45_{\pm 0.81}$	$21.69_{\pm 0.31}$	$23.48_{\pm 0.48}$	$\mathbf{36.28}_{\pm 0.63}$
	5-shot	$18.66_{\pm 0.05}$	$20.72_{\pm 0.24}$	$25.03_{\pm 0.81}$	$25.33_{\pm 0.33}$	$28.84_{\pm 0.79}$	$29.05_{\pm 0.19}$	$26.88_{\pm 0.74}$	$\mathbf{38.91}_{\pm 0.55}$
Pubmed	0-shot	$28.80_{\pm 0.73}$	$20.80_{\pm 0.43}$	$31.40_{\pm 0.19}$	$58.20_{\pm 0.97}$	$70.10_{\pm 0.21}$	$76.20_{\pm 0.92}$	$82.50_{\pm 0.46}$	$\mathbf{89.70}_{\pm 0.81}$
	1-shot	$32.61_{\pm 0.14}$	$29.14_{\pm 0.42}$	$35.25_{\pm 0.60}$	$62.40_{\pm 0.99}$	$70.55_{\pm 0.10}$	$76.47_{\pm 0.97}$	$84.69_{\pm 0.50}$	$\mathbf{90.35}_{\pm 0.69}$
	3-shot	$35.35_{\pm 0.04}$	$25.89_{\pm 0.91}$	$38.14_{\pm 0.17}$	$64.32_{\pm 0.50}$	$70.89_{\pm 0.13}$	$76.69_{\pm 0.06}$	$83.60_{\pm 0.45}$	$\mathbf{93.52}_{\pm 0.48}$
	5-shot	$38.57_{\pm 0.94}$	$32.50_{\pm 0.94}$	$40.12_{\pm 0.24}$	$65.44_{\pm 0.60}$	$71.35_{\pm 0.76}$	$76.95_{\pm 0.74}$	$85.76_{\pm 0.30}$	$\mathbf{94.15}_{\pm 0.76}$
Children	0-shot	$3.00_{\pm 0.62}$	$4.80_{\pm 0.17}$	$6.40_{\pm 0.73}$	$27.40_{\pm 0.05}$	$33.60_{\pm 0.77}$	$22.80_{\pm 0.69}$	$29.80_{\pm 0.99}$	$\mathbf{48.65}_{\pm 0.88}$
	1-shot	$11.39_{\pm 0.73}$	$9.04_{\pm 0.47}$	$13.03_{\pm 0.63}$	$29.85_{\pm 0.77}$	$38.64_{\pm 0.75}$	$34.14_{\pm 0.94}$	$33.32_{\pm 0.30}$	$\mathbf{52.84}_{\pm 1.02}$
	3-shot	$17.66_{\pm 0.98}$	$19.36_{\pm 0.36}$	$20.94_{\pm 0.70}$	$33.04_{\pm 0.11}$	$36.06_{\pm 1.02}$	$27.03_{\pm 0.74}$	$37.87_{\pm 0.75}$	$\mathbf{53.47}_{\pm 0.67}$
	5-shot	$20.74_{\pm 0.94}$	$15.13_{\pm 0.35}$	$17.25_{\pm 0.32}$	$36.23_{\pm 1.02}$	$41.72_{\pm 0.81}$	$31.40_{\pm 0.87}$	$35.57_{\pm 0.55}$	$\mathbf{55.28}_{\pm 0.71}$
History	0-shot	$6.50_{\pm 0.46}$	$16.80_{\pm 0.66}$	$5.20_{\pm 0.37}$	$29.10_{\pm 0.25}$	$31.90_{\pm 0.49}$	$20.80_{\pm 0.63}$	$49.80_{\pm 0.37}$	$\mathbf{51.45}_{\pm 0.59}$
	1-shot	$11.42_{\pm 0.95}$	$22.99_{\pm 0.96}$	$11.52_{\pm 0.68}$	$37.04_{\pm 0.08}$	$40.41_{\pm 0.20}$	$26.13_{\pm 1.01}$	$51.20_{\pm 0.10}$	$\mathbf{53.88}_{\pm 0.82}$
	3-shot	$18.41_{\pm 0.78}$	$25.75_{\pm 1.02}$	$16.07_{\pm 0.64}$	$31.94_{\pm 0.36}$	$34.29_{\pm 0.57}$	$29.01_{\pm 0.87}$	$\mathbf{55.39}_{\pm 0.08}$	$54.67_{\pm 0.94}$
	5-shot	$22.51_{\pm 0.07}$	$30.28_{\pm 0.90}$	$20.06_{\pm 0.93}$	$33.98_{\pm 0.16}$	$37.74_{\pm 0.48}$	$31.35_{\pm 0.46}$	$52.87_{\pm 0.42}$	$\mathbf{55.93}_{\pm 0.68}$
Photo	0-shot	$10.30_{\pm 0.57}$	$7.30_{\pm 0.44}$	$34.00_{\pm 0.64}$	$39.70_{\pm 0.22}$	$41.50_{\pm 0.46}$	$27.50_{\pm 0.10}$	$40.10_{\pm 1.00}$	$\mathbf{56.15}_{\pm 0.84}$
	1-shot	$15.93_{\pm 0.06}$	$18.80_{\pm 0.96}$	$37.47_{\pm 0.64}$	$42.71_{\pm 0.53}$	$43.15_{\pm 0.42}$	$31.24_{\pm 0.87}$	$42.02_{\pm 0.95}$	$\mathbf{57.92}_{\pm 0.66}$
	3-shot	$19.44_{\pm 0.28}$	$13.20_{\pm 0.71}$	$41.72_{\pm 0.56}$	$46.57_{\pm 0.95}$	$44.83_{\pm 0.86}$	$34.03_{\pm 0.45}$	$44.69_{\pm 0.30}$	$\mathbf{59.28}_{\pm 0.91}$
	5-shot	$24.53_{\pm 0.94}$	$22.90_{\pm 0.47}$	$39.12_{\pm 0.04}$	$44.73_{\pm 0.95}$	$47.10_{\pm 0.63}$	$36.24_{\pm 1.00}$	$46.21_{\pm 0.65}$	$\mathbf{60.74}_{\pm 0.77}$

Table 4: Link prediction AUC Score (cross-task) under different shot setting. The best results are in bold.

	Dataset K-shot	GraphGPT	GraphEdit	LLaGA	TEA-GLM	**GraphShaper**
Cora	0-shot	$50.20_{\pm 0.68}$	$58.20_{\pm 0.26}$	$58.80_{\pm 0.81}$	$55.30_{\pm 0.39}$	$\mathbf{60.38}_{\pm 0.31}$
	1-shot	$50.74_{\pm 0.12}$	$59.14_{\pm 0.65}$	$\mathbf{60.55}_{\pm 0.21}$	$56.00_{\pm 0.27}$	$59.01_{\pm 0.65}$
	3-shot	$52.21_{\pm 0.83}$	$60.01_{\pm 0.91}$	$59.51_{\pm 0.82}$	$57.10_{\pm 0.89}$	$\mathbf{61.99}_{\pm 0.88}$
	5-shot	$51.49_{\pm 0.34}$	$60.46_{\pm 0.38}$	$61.24_{\pm 0.55}$	$57.57_{\pm 0.49}$	$\mathbf{62.56}_{\pm 0.52}$
Pubmed	0-shot	$49.90_{\pm 0.98}$	$42.80_{\pm 0.43}$	$56.10_{\pm 0.18}$	$67.80_{\pm 0.97}$	$\mathbf{70.66}_{\pm 0.69}$
	1-shot	$50.39_{\pm 0.37}$	$44.47_{\pm 0.33}$	$56.91_{\pm 0.85}$	$69.08_{\pm 0.35}$	$\mathbf{74.13}_{\pm 0.71}$
	3-shot	$50.95_{\pm 0.42}$	$43.44_{\pm 0.20}$	$57.96_{\pm 0.96}$	$68.69_{\pm 0.72}$	$\mathbf{71.70}_{\pm 0.46}$
	5-shot	$52.00_{\pm 0.18}$	$45.01_{\pm 0.71}$	$58.97_{\pm 0.64}$	$69.59_{\pm 0.32}$	$\mathbf{75.25}_{\pm 0.91}$
Children	0-shot	$45.80_{\pm 0.33}$	$38.80_{\pm 0.64}$	$43.40_{\pm 0.97}$	$57.40_{\pm 0.27}$	$\mathbf{62.13}_{\pm 0.41}$
	1-shot	$46.60_{\pm 0.44}$	$39.47_{\pm 0.15}$	$44.17_{\pm 0.53}$	$57.88_{\pm 0.91}$	$\mathbf{65.29}_{\pm 0.35}$
	3-shot	$47.18_{\pm 0.37}$	$41.23_{\pm 0.96}$	$44.86_{\pm 1.02}$	$59.87_{\pm 0.53}$	$\mathbf{63.71}_{\pm 0.61}$
	5-shot	$47.99_{\pm 0.46}$	$40.12_{\pm 0.42}$	$45.81_{\pm 0.51}$	$58.88_{\pm 0.03}$	$\mathbf{66.82}_{\pm 0.42}$
History	0-shot	$28.20_{\pm 0.50}$	$42.20_{\pm 0.78}$	$48.10_{\pm 0.68}$	$53.10_{\pm 0.64}$	$\mathbf{65.37}_{\pm 0.88}$
	1-shot	$29.15_{\pm 0.22}$	$42.78_{\pm 0.55}$	$50.07_{\pm 0.21}$	$53.78_{\pm 0.07}$	$\mathbf{63.45}_{\pm 0.79}$
	3-shot	$30.14_{\pm 0.30}$	$43.58_{\pm 0.58}$	$49.07_{\pm 0.30}$	$55.26_{\pm 1.03}$	$\mathbf{66.91}_{\pm 0.28}$
	5-shot	$31.12_{\pm 0.63}$	$44.04_{\pm 0.06}$	$50.96_{\pm 1.01}$	$54.43_{\pm 0.73}$	$\mathbf{65.16}_{\pm 0.15}$

Table 5: Node description task using Sbert Score under different shot setting. The best results are in bold.

	Dataset K-shot	GraphGPT	GraphEdit	LLaGA	TEA-GLM	**GraphShaper**
Cora	0-shot	$64.11_{\pm 0.13}$	$61.09_{\pm 0.96}$	$86.72_{\pm 0.66}$	$70.20_{\pm 0.77}$	$\mathbf{90.82}_{\pm 0.39}$
	1-shot	$65.02_{\pm 0.99}$	$61.83_{\pm 1.00}$	$86.92_{\pm 0.43}$	$70.73_{\pm 0.31}$	$\mathbf{87.39}_{\pm 0.88}$
	3-shot	$65.75_{\pm 0.62}$	$62.27_{\pm 0.74}$	$87.17_{\pm 0.57}$	$71.64_{\pm 0.08}$	$\mathbf{91.11}_{\pm 1.09}$
	5-shot	$66.66_{\pm 0.88}$	$62.76_{\pm 0.77}$	$87.28_{\pm 0.46}$	$71.14_{\pm 0.47}$	$\mathbf{87.66}_{\pm 0.38}$
Pubmed	0-shot	$73.25_{\pm 0.84}$	$63.11_{\pm 0.84}$	$\mathbf{93.70}_{\pm 0.68}$	$85.41_{\pm 0.04}$	$89.70_{\pm 0.51}$
	1-shot	$73.72_{\pm 0.88}$	$64.69_{\pm 0.20}$	$92.58_{\pm 0.98}$	$85.66_{\pm 0.85}$	$\mathbf{94.50}_{\pm 0.86}$
	3-shot	$74.23_{\pm 0.73}$	$63.76_{\pm 0.53}$	$\mathbf{92.47}_{\pm 0.38}$	$85.84_{\pm 0.26}$	$90.04_{\pm 0.79}$
	5-shot	$74.82_{\pm 0.60}$	$65.31_{\pm 1.01}$	$89.96_{\pm 0.26}$	$86.11_{\pm 0.21}$	$\mathbf{93.86}_{\pm 0.49}$
Arxiv	0-shot	$58.92_{\pm 0.25}$	$55.67_{\pm 0.16}$	$74.64_{\pm 0.46}$	$60.04_{\pm 0.10}$	$\mathbf{79.61}_{\pm 0.44}$
	1-shot	$59.86_{\pm 0.47}$	$56.40_{\pm 0.19}$	$75.45_{\pm 0.30}$	$60.62_{\pm 0.72}$	$\mathbf{83.31}_{\pm 0.91}$
	3-shot	$61.33_{\pm 0.89}$	$57.19_{\pm 0.74}$	$75.89_{\pm 0.72}$	$61.65_{\pm 0.71}$	$\mathbf{80.45}_{\pm 0.68}$
	5-shot	$60.52_{\pm 0.11}$	$58.22_{\pm 0.71}$	$74.98_{\pm 1.01}$	$62.48_{\pm 0.14}$	$\mathbf{84.19}_{\pm 0.88}$
Products	0-shot	$66.45_{\pm 0.65}$	$62.01_{\pm 0.30}$	$83.18_{\pm 0.10}$	$69.77_{\pm 0.22}$	$\mathbf{90.45}_{\pm 0.76}$
	1-shot	$67.35_{\pm 0.90}$	$63.67_{\pm 0.55}$	$83.39_{\pm 0.21}$	$70.23_{\pm 0.35}$	$\mathbf{87.19}_{\pm 0.45}$
	3-shot	$68.12_{\pm 0.31}$	$62.92_{\pm 0.74}$	$83.57_{\pm 0.59}$	$71.29_{\pm 0.88}$	$\mathbf{90.60}_{\pm 0.48}$
	5-shot	$68.72_{\pm 0.05}$	$64.38_{\pm 1.01}$	$83.84_{\pm 0.66}$	$70.73_{\pm 0.67}$	$\mathbf{87.37}_{\pm 0.52}$

5.4 Main Result

Zero-shot and few-shot transfer performance demonstrates our superiority across diverse domains. Table 3 shows GraphShaper achieves **29.80%** accuracy on Cora and **89.70%** on Pubmed in zero-shot settings, outperforming the strongest baseline TEA-GLM by substantial margins. Table 4 validates consistent link prediction improvements with **70.66%** AUC on Pubmed. Table 5 demonstrates superior node description quality using SBERT scores above **90%** on both Cora and Products. Table 6 further confirms cross-dataset generalization with **5.7%** improvement over RiemannGFM when transferring from Arxiv and PubMed. These results collectively demonstrate that adaptive geometric fusion enables robust knowledge transfer without requiring labeled target data.

Table 6: Generalization performance of GraphShaper on zero-shot node classification across multi-dataset transfer settings.

Model	(Arxiv+Pubmed+Cora) → Products	(Arxiv+Cora) → PubMed	(Arxiv+Cora) → Arxiv	(Arxiv+PubMed) → Cora	(Arxiv+PubMed) → Arxiv
LLaGA	13.8%	42.1%	48.2%	16.4%	46.7%
TEA-GLM	9.1%	38.6%	50.3%	21.9%	54.7%
GraphCLIP	11.7%	46.9%	55.7%	19.3%	58.4%
RiemannGFM	13.4%	43.8%	52.4%	25.1%	67.9%
GraphShaper	19.7%(4.4% ↑)	51.3%(4.4% ↑)	60.2%(4.5% ↑)	27.8%(2.7% ↑)	73.6%(5.7% ↑)

Table 7: Ablation study results for each architecture component. L_{MHS} denotes expert diversity loss. Δ represents the performance gap compared to the full model.

Architecture Components					Cora		Children	
Multi-Geo	Hyper	Sphere	Gating	L_{MHS}	Acc	Δ	Acc	Δ
✓	✓	✓	✓	✓	**29.80**	-	**48.65**	-
					19.20	−10.60	29.80	−18.85
✓		✓	✓	✓	24.50	−5.30	46.10	−2.55
✓	✓		✓	✓	27.90	−1.90	45.30	−3.35
✓	✓	✓		✓	22.10	−7.70	42.70	−5.95
✓	✓	✓	✓		26.50	−3.30	46.90	−1.75

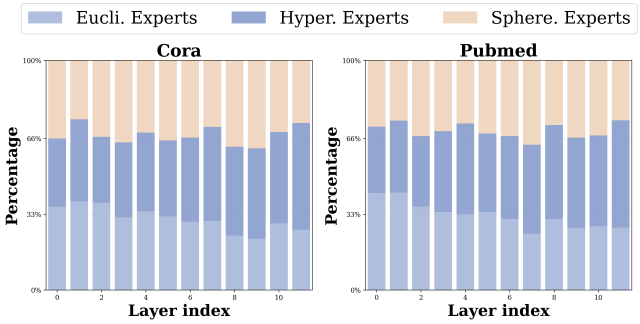


Figure 3: Geometric expert activation distribution across all transformer layers.

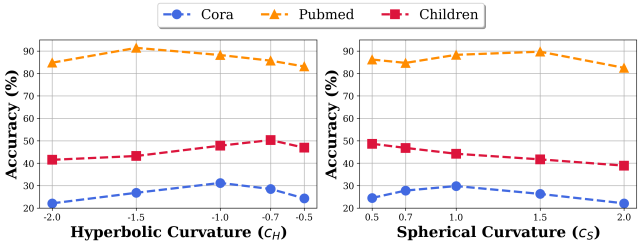


Figure 4: Impact of Curvature Parameters on Model Performance.

5.5 Model Analysis

Geometric expert activation patterns reveal adaptive specialization across network layers. Figure 3 illustrates expert distributions where hyperbolic experts dominate with approximately **60–70%** activation, reflecting hierarchical citation structures in academic networks. Spherical experts maintain **20–30%** activation for cyclic patterns while Euclidean experts contribute **10–20%** for local connectivity. The consistent distribution across layers indicates stable geometric specialization throughout network depth. Notably, Pubmed shows higher hyperbolic activation than Cora, suggesting stronger hierarchical organization in medical literature. The balanced utilization validates multi-geometric fusion over single-space encoding.

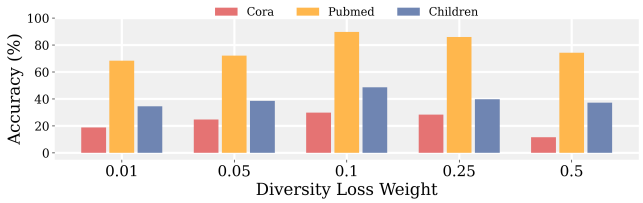


Figure 5: Impact of Expert Diversity Loss Weight.

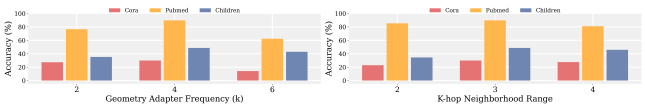


Figure 6: Performance and efficiency trade-off of geometry adapter frequency and analysis of K-hop neighborhood sampling range.

5.6 Hyper-parameter Analysis

Curvature parameters and diversity loss weight critically influence model performance. Figure 4 shows optimal hyperbolic curvature around **-1.0** and spherical curvature around **1.0** across datasets, with Pubmed maintaining highest accuracy due to its hierarchical structure. Extreme values degrade performance by misrepresenting geometric properties. Figure 5 reveals moderate diversity loss weights between **0.05** and **0.1** yield optimal performance, while excessive weights above **0.25** over-constrain expert parameters. Figure 6 demonstrates geometry adapter frequency of **4 layers** and **k-hop range of 3** provide the best balance between performance and efficiency. These analyses confirm careful geometric parameter calibration enables effective structural encoding.

5.7 Ablation Study

Each architectural component contributes significantly through specialized geometric modeling. Table 7 shows removing all geometry-aware components reduces Cora accuracy from **29.80%** to **19.20%**

and Children accuracy from **48.65%** to **29.80%**. Removing hyperbolic experts alone causes substantial drops, demonstrating their critical role in hierarchical encoding. Eliminating spherical experts reduces cyclic pattern modeling capability. Removing the gating network leads to major degradation, validating adaptive fusion over static combination. Disabling diversity loss confirms orthogonality constraints prevent parameter redundancy. The cumulative effect exceeds individual contributions, indicating synergistic interactions between geometric experts and fusion mechanisms. These results demonstrate geometry-aware design is essential for structural heterogeneity.

6 Conclusion

This work addresses geometric representation conflicts in graph foundation models caused by encoding diverse structural patterns in uniform spaces. Current methods using single Euclidean spaces create trade-offs that distort structural information, especially problematic in cross-domain transfer. GraphShaper resolves this through adaptive geometric fusion, where each node dynamically integrates multiple geometric properties according to local context. Strong zero-shot and few-shot results validate this geometry-aware approach.

References

[1] William Brannon, Suyash Fulay, Hang Jiang, Wonjune Kang, Brandon Roy, Jad Kabbara, and Deb Roy. 2023. ConGraT: Self-Supervised Contrastive Pretraining for Joint Graph and Text Embeddings. *arXiv preprint arXiv:2305.14321* (2023).

[2] William Brannon, Suyash Fulay, Hang Jiang, Wonjune Kang, Brandon Roy, Jad Kabbara, and Deb Roy. 2023. Congrat: Self-supervised contrastive pretraining for joint graph and text embeddings. *arXiv preprint arXiv:2305.14321* (2023).

[3] Yukun Cao, Shuo Han, Zengyi Gao, Zezhong Ding, Xike Xie, and S. Kevin Zhou. 2024. GraphInsight: Unlocking Insights in Large Language Models for Graph Structure Understanding. *arXiv preprint arXiv:2409.03258* (2024).

[4] Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. GraphLLM: Boosting Graph Reasoning Ability of Large Language Model. *arXiv preprint arXiv:2310.05845* (2023).

[5] Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. 2024. GraphWiz: An Instruction-Following Language Model for Graph Problems. *arXiv preprint arXiv:2402.16029* (2024).

[6] Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. 2024. LLaGA: Large Language and Graph Assistant. *arXiv preprint arXiv:2402.08170* (2024).

[7] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2023. Exploring the Potential of Large Language Models (LLMs) in Learning on Graph. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*. https://openreview.net/forum?id=ScNNo7v4t0

[8] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2023. Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs. *arXiv preprint arXiv:2307.03393* (2023).

[9] Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, et al. 2024. Text-space Graph Foundation Models: Comprehensive Benchmarks and New Insights. *arXiv preprint arXiv:2406.10727* (2024).

[10] Zhikai Chen, Haitao Mao, Hongzhi Wen, et al. 2023. Label-free Node Classification on Graphs with Large Language Models (LLMs). In *The Twelfth International Conference on Learning Representations*.

[11] Wenqi Fan, Shijie Ma, Zhihua Zhong, Xiaorui Xie, Yun Liu, Yiwei Chen, Jiatu Zhao, Yuchen Peng, Jianxin Zhang, Yuchen Wang, et al. 2024. Graph Machine Learning in the Era of Large Language Models (LLMs). *arXiv preprint arXiv:2404.14928* (2024).

[12] Jiayan Guo, Lun Du, and Hengyu Liu. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066* (2023).

[13] Jiaxin Guo, Ziyu Wang, Zhihui Zhao, et al. 2023. GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking. *arXiv preprint arXiv:2305.15066* (2023).

[14] Zirui Guo, Lianghao Xia, Yanhua Yu, Yuling Wang, Zixuan Yang, Wei Wei, Liang Pang, Tat-Seng Chua, and Chao Huang. 2024. GraphEdit: Large Language Models for Graph Structure Learning. *arXiv preprint arXiv:2402.15183* (2024).

[15] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning*. 11 pages.

[16] Xiaoxin He, Xavier Bresson, Thomas Laurent, and Bryan Hooi. 2023. Explanations as Features: LLM-Based Features for Text-Attributed Graphs. *arXiv preprint arXiv:2305.19523* (2023).

[17] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. *arXiv preprint arXiv:2305.19523* (2023).

[18] Yufei He and Bryan Hooi. 2024. UniGraph: Learning a Cross-Domain Graph Foundation Model From Natural Language. *arXiv preprint arXiv:2402.13630* (2024).

[19] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[21] Xunkai Li et al. 2024. Graph Learning in the Era of LLMs: A Survey from the Perspective of Data, Models, and Tasks. *arXiv preprint arXiv:2412.12456* (2024).

[22] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2024. A Survey of Graph Meets Large Language Model: Progress and Future Directions. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 8123–8131. doi:10.24963/ijcai.2024/898 Survey Track.

[23] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 1725–1735. doi:10.1145/3637528.3671982

[24] Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor Wai Kin Chan, and Jia Li. 2024. GLBench: A Comprehensive Benchmark for Graph with Large Language Models. *arXiv preprint arXiv:2407.07457* (2024).

[25] Chang Liu and Bo Wu. 2023. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224* (2023).

[26] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, and Dacheng Tao. 2023. One for All: Towards Training One Graph Model for All Classification Tasks. *arXiv preprint arXiv:2310.00149* (2023).

[27] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2023*. 417–428. https://doi.org/10.1145/3543507.3583386

[28] Zihan Luo, Xiran Song, Hong Huang, Jianxun Lian, Chenhao Zhang, Jinqi Jiang, Xing Xie, and Hai Jin. 2024. GraphInstruct: Empowering Large Language Models with Graph Understanding and Reasoning Capability. *arXiv preprint arXiv:2403.04483* (2024).

[29] Zheqi Lv, Tianyu Zhan, Wenjie Wang, Xinyu Lin, Shengyu Zhang, Wenqiao Zhang, Jiwei Li, Kun Kuang, and Fei Wu. 2025. Collaboration of Large Language Models and Small Recommendation Models for Device-Cloud Recommendation. *arXiv preprint arXiv:2501.05647* (2025).

[30] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2015), 43–52.

[31] Bryan Perozzi et al. 2024. Let Your Graph Do the Talking: Encoding Structured Data for LLMs. *arXiv preprint arXiv:2402.05862* (2024).

[32] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160. https://doi.org/10.1145/3394486.3403168

[33] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3464–3475.

[34] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. *AI Magazine* 29, 3 (2008), 93–106.

[35] Li Sun et al. 2025. RiemannGFM: Learning a Graph Foundation Model from Riemannian Geometry. *arXiv preprint arXiv:2502.03251* (2025).

[36] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. GraphGPT: Graph Instruction Tuning for Large Language Models. *arXiv preprint arXiv:2310.13023* (2023).

[37] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rklz9iAcKQ

[38] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings. *arXiv preprint arXiv:2408.14512* (2024).

[39] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[40] Heng Wang, Shangbin Feng, Tianxing He, et al. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems* (2024).

[41] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems* 36 (2024).

[42] Haishuai Wang, Yang Gao, Xin Zheng, et al. 2023. Graph neural architecture search with gpt-4. *arXiv preprint arXiv:2310.01436* (2023).

[43] Xiaotang Wang, Yun Zhu, Haizhou Shi, Yongchao Liu, and Chuntao Hong. 2024. UniGAP: A Universal and Adaptive Graph Upsampling Approach to Mitigate Over-Smoothing in Node Classification Tasks. arXiv:2407.19420 [cs.LG] https://arxiv.org/abs/2407.19420

[44] Zhihao Wen and Yuan Fang. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 506–516.

[45] Qitian Wu, Wentao Liu, Hengrui Yang, Jing Zhao, Fuzhen Zhou, and Jianyong Wang. 2023. NodeFormer: A Scalable Graph Structure Learning Transformer for Node Classification. *arXiv preprint arXiv:2306.08385* (2023).

[46] Rui Yang, Jiahao Zhu, Jianping Man, Li Fang, and Yi Zhou. 2024. Exploiting Large Language Models Capabilities for Question Answer-Driven Knowledge Graph Completion Across Static and Temporal Domains. *arXiv preprint arXiv:2408.10819* (2024).

[47] Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. *arXiv preprint arXiv:1603.08861* (2016).

[48] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134* (2023).

[49] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134* (2023).

[50] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *Advances in Neural Information Processing Systems*. 28877–28888. https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf

[51] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph Contrastive Learning Automated. In *ICML*. https://arxiv.org/abs/2106.07594

[52] Jianxiang Yu et al. 2023. Leveraging Large Language Models for Node Generation in Few-Shot Learning on Text-Attributed Graphs. *arXiv preprint arXiv:2310.09872* (2023).

[53] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, et al. 2023. Empower text-attributed graphs learning with large language models (llms). *arXiv preprint arXiv:2310.09872* (2023).

[54] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM Web Conference 2023*.

[55] Zihao Zhang, Xunkai Li, et al. 2025. Toward General and Robust LLM-enhanced Text-attributed Graph Learning. *arXiv preprint arXiv:2504.02343* (2025).

[56] Haiteng Zhao, Shengchao Liu, Ma Chang, et al. 2024. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in Neural Information Processing Systems* (2024).

[57] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2022. Learning on Large-scale Text-attributed Graphs via Variational Inference. In *Proc. of ICLR*.

[58] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089* (2023).

[59] Ziwen Zhao, Yifei Su, Yu Li, Yue Zou, Ruoyu Li, and Ruiming Zhang. 2024. A Survey on Self-Supervised Graph Foundation Models: Knowledge-Based Perspective. *arXiv preprint arXiv:2403.16137* (2024).

[60] Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. 2025. LLM as GNN: Graph Vocabulary Learning for Text-Attributed Graph Foundation Models. *arXiv preprint arXiv:2503.03313* (2025).

[61] Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. 2024. GraphCLIP: Enhancing Transferability in Graph Foundation Models for Text-Attributed Graphs. *arXiv preprint arXiv:2410.10329* (2024).

[62] Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient Tuning and Inference for Large Language Models on Textual Graphs. *arXiv preprint arXiv:2401.15569* (2024).

[63] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).

[64] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *Proceedings of the Web Conference 2021*. 2069–2080. https://doi.org/10.1145/3442381.3449802