GASE: Graph-Aware Semantic Embedding Learning with Frozen LLMs for Text-Attributed Graphs

Mingqiang Ding[1], **Jianjun Li**[1*], **Wenqi Yang**[1], **Zhibo Zhang**[1], **Yushen Fang**[1]
[1] School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
{mingqianding, jianjunli, yangwenqi, zhibo_zhang, yushen_fang}@hust.edu.cn

Abstract

Large Language Models (LLMs) have shown strong potential for text-attributed graph (TAG) learning, yet effectively integrating LLM semantics with graph structural information remains challenging. Embeddings obtained from frozen LLMs lack topology awareness, while fine-tuning LLMs is often computationally expensive. Moreover, LLM embeddings are high-dimensional, and naively reducing dimensionality tends to destroy semantics. To address these issues, we propose **GASE**, a framework for learning Graph-Aware Semantic Embeddings using frozen LLMs. GASE consists of two key stages: First, we introduce a Training-Free Structure-Aware Semantic Extraction (TSSE) module. Through inter-layer semantic feedback and progressive masked attention, it efficiently compresses and propagates semantic context from neighboring nodes without updating LLM parameters. Second, we propose a Subspace Decomposition and Structural Injection (SDSI) strategy. Embeddings obtained from TSSE are decomposed into a semantic-rich subspace and a structural injection subspace, and structural signals are injected into the latter, which preserves original semantics while integrating graph information. Experiments demonstrate that GASE outperforms state-of-the-art baselines on node classification and achieves a 5× speedup over fine-tuning-based methods.

1 Introduction

Text-Attributed Graphs (TAGs), which encompass rich textual attributes and structural relationships, are ubiquitous in real-world scenarios such as academic citation networks (Wang et al., 2020), social media (Hu et al., 2025), and knowledge graphs (Wen et al., 2024). Effectively integrating textual and structural information to learn accurate node representations remains a central and ongoing challenge in graph machine learning (Zhou et al., 2025a; Zhang et al., 2024). Recently, the emergence of large language models (LLMs) has opened new pathways for addressing this challenge, attracting growing research efforts in LLM-enhanced TAG learning (Chen et al., 2024; Wang et al., 2025).

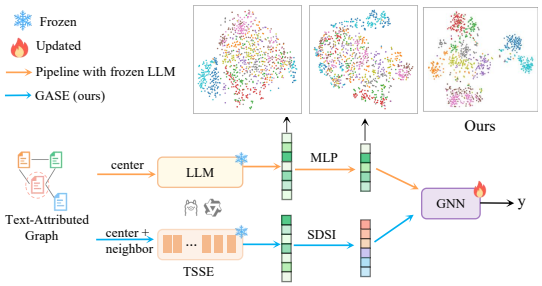


Figure 1: Comparison of the existing frozen-LLM-based pipeline for TAGs and our proposed method.

Existing LLM-enhanced TAG learning approaches (Zhang et al., 2025; Fang et al., 2024) can be classified into two main categories. The first treats LLMs as text augmenters to generate additional node descriptions. While this can enrich textual information, it often introduces potential noise (Ding et al., 2024) and high computational overhead. The second category uses LLMs as embedding generators, which can be further subdivided into two strategies: fine-tuning LLMs or directly using frozen LLMs to generate embeddings. Although fine-tuning can adapt language representations through task-specific supervision, it incurs prohibitively high computational and memory costs (Wang et al., 2024), severely limiting scalability on large graphs. Consequently, many studies have turned to frozen LLMs as semantic encoders for TAG learning (Liu et al., 2024; Zhou et al., 2025b).

Nevertheless, existing frozen-LLM-based methods still face two key challenges. First, embeddings directly generated by LLMs are inherently structure-agnostic (Yang et al., 2025b), lacking explicit neighborhood semantics and showing limited discriminability (see Fig. 1). Simply concatenating

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31073–31086

neighbor texts to inject structural context results in excessively long input sequences, which is impractical for large graphs. Second, LLM embeddings are typically high-dimensional and anisotropic, whereas Graph Neural Networks (GNNs) operate in low-dimensional structural spaces (Hu et al., 2024). Directly projecting LLM embeddings into the GNN space via MLPs often disrupts the original semantic geometry (Yang et al., 2025a) (see Fig. 1), leading to low-dimensional representations with uneven distributions and semantic distortion, thereby limiting the subsequent GNN modeling.

To overcome these challenges, we propose GASE, a fine-tuning-free framework based on frozen LLMs for learning Graph-Aware Semantic Embeddings. As depicted in Fig. 1, GASE operates in two stages. First, we introduce a Training-Free Structure-Aware Semantic Extraction (TSSE) module, a plug-and-play component leveraging inter-layer semantic feedback and progressively masked attention that can be attached to frozen LLMs to enable neighbor-aware semantic propagation without parameter updates. Second, we propose a Subspace Decomposition and Structural Injection (SDSI) strategy. Embeddings obtained from TSSE are decomposed into a semantic-rich subspace and a structural injection subspace. Graph signals are then injected into the structural subspace, preserving the original semantic geometry while integrating topological information. A GNN is subsequently applied to the fused representation. Our contributions are outlined as follows:

- We propose a novel, fine-tuning-free framework that equips frozen LLMs with explicit structural awareness for TAG learning.
- We design a plug-and-play module using inter-layer semantic feedback and progressively masked attention to enable neighbor-aware semantic aggregation within a frozen LLM.
- We propose a novel subspace-based dimensionality reduction strategy that integrates graph structural signals while preserving the semantic information encoded by the LLM.
- Extensive experiments show that GASE outperforms state-of-the-art baselines on node classification tasks, achieving superior performance to fine-tuning-based methods with significantly lower computational cost.

2 Related work

Traditional LM Fine-Tuning Methods in TAGs With the advent of LMs such as BERT (Devlin et al., 2019), fine-tuning LMs on downstream tasks has become a prevalent strategy for TAG learning (Duan et al., 2023; Zhao et al., 2024; Liu et al., 2025). For example, GLEM (Zhao et al., 2023) introduces a variational Expectation-Maximization framework, where LMs and GNNs are alternately updated. CTGL (Zhou et al., 2025a) proposes a coupled fine-tuning framework that jointly optimizes LMs and GNNs in parallel. UniGLM (Fang et al., 2025) fine-tunes LMs using a domain-aware contrastive learning objective that unifies structural heterogeneity across diverse domains.

LLM-Enhanced Methods for TAGs The first paradigm regards LLMs as textual enhancers (Yang et al., 2024b; Yu et al., 2025). TAPE (He et al., 2024) leverages LLMs to craft explanatory text for each node. Although these methods enrich textual information, they rely heavily on the quality of the LLM outputs, which can be noisy. The second paradigm treats LLMs as embedding generators, which can be further subdivided into two strategies: fine-tuning LLMs or directly using frozen LLMs. LanSAGNN (Li et al., 2025) proposes a dual-layer LLM fine-tuning architecture to align LLMs' outputs with graph tasks. GraphBridge (Wang et al., 2024) employs LoRA for LLM fine-tuning on node text together with neighbor context. Such fine-tuning methods incur high computational costs. OFA (Liu et al., 2024) and GraphAdapter (Huang et al., 2024) directly use frozen LLMs to generate embeddings and use MLPs to project the embeddings into low-dimensional space required by GNNs. However, this straightforward mapping approach inevitably leads to information loss (BehnamGhader et al., 2024).

3 Preliminary

We consider a Text-Attributed Graph (TAG) defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ the set of edges, and $\mathcal{T} = \{T_i\}$ denotes the textual attributes associated with nodes. The objective of TAG representation learning is to obtain node fused embeddings e_{fused} that capture both semantic information from text and structural relationships within the graph, so as to support downstream graph-related tasks such as node classification and link prediction. In practice, LLMs

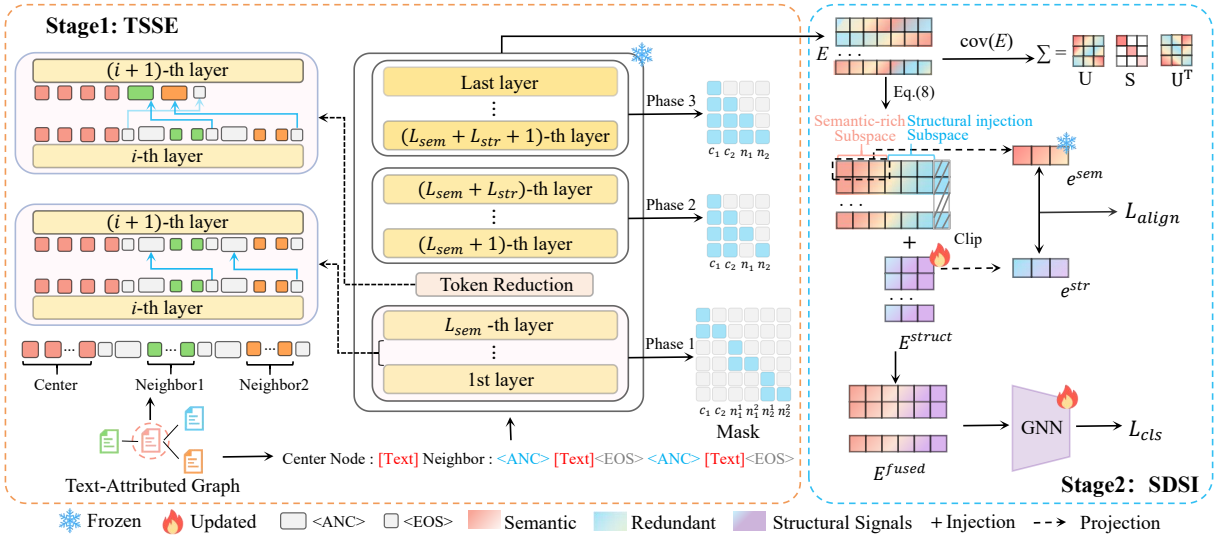


Figure 2: Overview of the proposed GASE.

provide strong semantic encoding ability but lack structural awareness, motivating our method to fuse structure into frozen LLM embeddings effectively.

4 Methodology

The overall architecture of GASE is illustrated in Fig. 2. The framework operates in two sequential stages: (1) Training-Free Structure-Aware Semantic Extraction (TSSE), which produces compact, graph-aware node semantic representations without fine-tuning the LLM; followed by (2) Subspace Decomposition and Structural Injection (SDSI), which fuses structural signals while preserving original semantic geometry, yielding discriminative node representations for downstream tasks.

4.1 Training-Free Structure-Aware Semantic Extraction (TSSE)

This stage employs a frozen decoder-only LLM as a feature extractor to derive graph-aware node representations while retaining its pretrained semantic capacity. Without updating any model parameters, we introduce an Inter-layer Semantic Feedback (ISF) mechanism coupled with a progressively attention masking strategy to modulate information flow during forward inference. This design enables zero-shot, parameter-free neighbor compression and semantic aggregation in a structured and controllable manner.

4.1.1 Input Construction

Given a center node v_i and its sampled neighbor set $\mathcal{N}(i) = \{v_{j_1}, \ldots, v_{j_m}\}$, we first construct an independent input sequence for each neighbor. For a neighbor text $T_j = [t_1, \ldots, t_{l_j}]$, we prepend a **Semantic Anchor Position** <ANC> to form the neighbor sequence as $S_j = [\texttt{<ANC>}; T_j]$, where the anchor <ANC> is not a new vocabulary token. Instead, it is implemented as a designated anchor position in the input sequence that reuses an existing special token embedding (e.g., the BOS token). All neighbor sequences are then concatenated with the main node text to form the complete input as $S_i = [T_i; S_{j_1}; \ldots; S_{j_m}]$.

4.1.2 Inter-layer Semantic Feedback

In the standard causal masking scheme, the anchor token <ANC> cannot directly attend to subsequent tokens within the same layer. To overcome this limitation without updating model parameters, we establish an implicit inter-layer feedback pathway across the first L_{sem} layers. For each neighbor sequence S_j, we consider its corresponding end-of-sequence token, denoted as $\texttt{<EOS>}_j$, which naturally aggregates the layer-wise global semantics under causal attention. We also denote the end token of the center-node text T_i as $\texttt{<EOS>}_i$.

Formally, let $\mathbf{h}^{(l)}_{\texttt{<ANC>}_j}$ and $\mathbf{h}^{(l)}_{\texttt{<EOS>}_j}$ denote the hidden states of the anchor position and the end-of-sequence token for neighbor v_j at layer l, respectively. The inter-layer feedback rule is defined as:

$$\mathbf{h}^{(l+1)}_{\texttt{<ANC>}_j} \leftarrow \mathbf{h}^{(l)}_{\texttt{<EOS>}_j}, \quad 1 \le l < L_{sem}. \tag{1}$$

Through this inter-layer feedback pathway, each anchor token $\texttt{<ANC>}_j$ progressively accumulates semantics of its corresponding neighbor. After L_{sem} layers, the hidden state of each anchor is extracted as the compressed neighbor representation: $r_j =$

$\mathbf{h}^{(L_{sem})}_{\texttt{<ANC>}_j}$. After obtaining $\mathcal{R}_i = \{r_{j_1}, \ldots, r_{j_m}\}$ where each r_{j_k} is a continuous hidden-state vector, we construct the input to layer $L_{\text{sem}} + 1$ by keeping the center-node sequence and inserting the compressed neighbor representations immediately before the center-node end token:

$$\tilde{\mathbf{H}}_i = [\mathbf{H}(T_i);\, r_{j_1};\, \ldots;\, r_{j_m};\, \mathbf{h}^{(L_{sem})}_{\texttt{<EOS>}_i}], \quad (2)$$

where $\mathbf{H}(T_i)$ denotes the hidden states of the center-node text at the output of the L_{sem}-th layer excluding its end token. In practice, we allocate m placeholder positions and overwrite their hidden states at layer L_{sem} with $\{r_{j_k}\}$, without introducing any trainable parameters.

4.1.3 Progressively Structure-Aware Attention

To precisely regulate information flow, we propose a progressively attention mask strategy, where the masking pattern evolves with the network depth. The process is divided into three phases, each controlled by a depth-dependent attention mask $\mathbf{M}^{(l)}$. The attention score at layer l is computed as:

$$A_{pq} = \text{Softmax}\left(\frac{\mathbf{q}_p \mathbf{k}_q^\top}{\sqrt{d}} + \mathbf{M}^{\text{causal}}_{pq} + \mathbf{M}^{\text{struct}}_{pq}\right). \quad (3)$$

$\mathbf{M}^{causal}_{pq}$ is the standard lower-triangular causal mask and $\mathbf{M}^{struct}$ is defined piecewise as follows:

1) Semantic Accumulation Phase ($1 \le l \le L_{sem}$). Aligned with the ISF mechanism, we apply a segment-isolation mask to ensure each text segment is processed independently, preventing irrelevant neighbor tokens from interfering with semantic consolidation of each node:

$$\mathbf{M}^{\text{struct}}_{pq} = \begin{cases} 0, & v(p) = v(q), \\ -\infty, & v(p) \neq v(q). \end{cases} \quad (4)$$

where $v(p)$ denotes the node to which token p belongs. After the first L_{sem} layers, we reconstruct the hidden-state sequence following Eq. (2).

2) Structural Propagation Phase ($L_{sem} < l \le L_{sem} + L_{str}$). We adopt a star-topology mask that allows each neighbor summary to attend to the center node, while blocking interactions among neighbors, thereby reducing computational complexity and mitigating cross-node semantic interference:

$$\mathbf{M}^{\text{struct}}_{pq} = \begin{cases} -\infty, & p \in \mathcal{N},\ q \in \mathcal{N},\ q \neq p, \\ 0, & \text{otherwise}. \end{cases} \quad (5)$$

where $\mathcal{N}$ denotes the set of summary tokens corresponding to the neighboring nodes.

3) Global Integration Phase ($l > L_{sem} + L_{str}$). In deeper layers, all additional structural constraints are removed, restoring standard causal attention of the decoder-only LLM. The model then performs autoregressive integration of the previously formed semantic and structurally modulated representations: $\mathbf{M}^{\text{struct}}_{pq} = \mathbf{0}$. After L layers of transformation, we extract the hidden state of the center-node end token $\texttt{<EOS>}_i$, which is placed at the end of the entire input sequence, as the final representation of node v_i, which fuses the intrinsic semantic features $S(v_i)$ and the neighbor semantics, serving as the input to Stage II.

Efficient Caching. Since TSSE employs a frozen LLM and performs training-free node-wise semantic encoding, the compressed neighbor representations obtained in the semantic accumulation phase remain invariant across training iterations. We therefore cache these semantic summaries and reuse them for different center nodes and epochs. This caching strategy greatly reduces redundant forward passes through the LLM, improving efficiency while preserving semantic fidelity.

4.2 Subspace Decomposition and Structural Injection (SDSI)

While TSSE produces high-quality node embeddings enriched with local context, directly applying these embeddings to graph tasks encounters two primary challenges: (1) the absence of explicit, learnable structural signals, and (2) the semantic distortion that can arise from the dimensionality mismatch between semantic and structural spaces. To address these issues, we propose a subspace decomposition and structural injection strategy. This approach decomposes the LLM embeddings into a semantic-rich subspace and a complementary structural injection subspace, enabling controlled integration of topological information without corrupting the pretrained semantic knowledge.

4.2.1 Semantic Space Decomposition

We begin with the LLM embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times d_l}$ obtained from TSSE, where N is the number of nodes and d_l is the embedding dimension. The covariance matrix of $\mathbf{E}$ is computed as:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{e}_i - \mu)(\mathbf{e}_i - \mu)^\top, \quad \mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{e}_i. \quad (6)$$

To preserve semantics and avoid degradation from direct dimensionality reduction, we perform

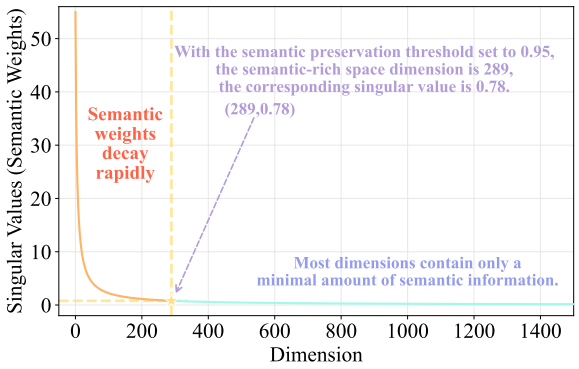


Figure 3: Eigenvalue spectrum of the embeddings.

eigendecomposition on the covariance matrix:

$$\Sigma = \mathbf{U}\mathbf{S}\mathbf{U}^\top, \tag{7}$$

where $\mathbf{S} = \mathrm{diag}(s_1^2, \ldots, s_{d_l}^2)$ contains the eigenvalues of Σ, and each column $\mathbf{u}_i$ of $\mathbf{U}$ is the corresponding orthonormal eigenvector satisfying $\Sigma\mathbf{u}_i = s_i^2\mathbf{u}_i$. Each eigenvector $\mathbf{u}_i$ spans a one-dimensional semantic subspace: $V_i = \mathrm{span}(\mathbf{u}_i) \subset \mathbb{R}^{d_l}$. A larger s_i indicates that the corresponding subspace captures a higher proportion of semantic variance in the embedding space.

4.2.2 Subspace Partitioning

As shown in Figure 3, the eigenvalue spectrum of Σ decays rapidly, indicating that most semantic variance concentrates in a few dominant principal subspaces. Accordingly, we partition the semantic space into two complementary subspaces:

Semantic-Rich Subspace: $\mathcal{V}_1 = \{V_1, \ldots, V_{d_s}\}$ consists of the top-d_s subspaces associated with the largest eigenvalues, which capture the majority of semantic content. The dimension d_s is determined as $d_s = \min\left\{k \;\middle|\; \frac{\sum_{i=1}^{k} s_i^2}{\sum_{j=1}^{d_l} s_j^2} \geq \tau\right\}$. where τ (typically set to 0.95) controls the proportion of preserved semantic variance.

Structural Injection Subspace (Semantic-Sparse): $\mathcal{V}_2 = \{V_{d_s+1}, \ldots, V_{d_l}\}$ comprises the remaining $d_l - d_s$ low-variance subspaces associated with near-zero eigenvalues ($s_i^2 \approx 0$), containing negligible semantic content. This subspace provides a suitable, low-interference space for injecting structural signals.

This optimal partitioning is theoretically guaranteed by the Eckart-Young-Mirsky theorem (Eckart and Young, 1936), which ensures minimal reconstruction error for any rank-d_s approximation.

4.2.3 Subspace Clipping and Normalization

Given the inherent low-rank structure of the semantic space, the dimensionality of the structural injection subspace is typically much larger than necessary, introducing redundancy. To mitigate this, we clip the structural injection subspace by retaining only the first d_n principal directions for efficient structural embedding learning. Furthermore, to address the anisotropic distribution of semantic variance across principal directions, we apply spectral normalization via variance scaling. The final normalized embedding is computed as:

$$\mathbf{E}_{\text{node}} = (\mathbf{E} - \mu)\,\mathbf{U}[:, : d_s + d_n]\,\mathbf{S}^{-\frac{1}{2}}[: d_s + d_n, : d_s + d_n], \tag{8}$$

where $\mathbf{E}_{\text{node}}$ is a joint representation that preserves semantic expressiveness in the first d_s dimensions while pre-allocating the remaining d_n dimensions for subsequent structural embedding integration.

4.2.4 Structural Information Injection

In this step, we keep the normalized semantic-rich subspace frozen and learn structural embeddings exclusively within the pruned structural injection subspace. Specifically, we introduce a structural embedding matrix $\mathbf{E}_{\text{struct}} \in \mathbb{R}^{N \times d_n}$, whose parameters are randomly initialized and jointly optimized with the GNN parameters. The learned structural embeddings are injected into the last d_n dimensions to obtain the fused representation:

$$\mathbf{E}_{\text{fused}} = \mathbf{E}_{\text{node}} + [\mathbf{0} \in \mathbb{R}^{N \times d_s},\ \mathbf{E}_{\text{struct}}]. \tag{9}$$

A GNN then performs multi-layer message passing over $\mathbf{E}_{\text{fused}}$ to aggregate structural features from neighbors:

$$\mathbf{E}_{\text{final}} = \mathbf{H}^{(L)} = \mathrm{GNN}(\mathbf{E}_{\text{fused}}, \mathbf{A}). \tag{10}$$

where $\mathbf{A}$ is the graph adjacency matrix. The resulting node representations achieve effective dimensionality reduction while holistically fusing deep semantic and structural information.

To enhance consistency between the two subspaces, we introduce a cross-space alignment loss. For a batch of nodes, we extract their semantic embeddings e^{sem} from the semantic-rich subspace and structural embeddings e^{str} after structural injection. Both embeddings are projected into a shared latent space via projection heads, yielding $\hat{e}^{\text{sem}}$ and $\hat{e}^{\text{str}}$. The cross-space alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\phi(\hat{e}_i^{\text{sem}}, \hat{e}_i^{\text{str}})/\gamma\right)}{\sum_{j=1}^{B} \exp\left(\phi(\hat{e}_i^{\text{sem}}, \hat{e}_j^{\text{str}})/\gamma\right)}, \tag{11}$$

where $\phi(\cdot,\cdot)$ denotes the cosine similarity and γ denotes the temperature parameter.

4.3 Optimization

For node classification, we obtain the fused representation $\mathbf{h}_v$ for each node v from $\mathbf{E}_{\text{final}}$ to predict the label as follows: $\hat{y} = \text{softmax}(\text{MLP}(\mathbf{h}_v))$. We then employ standard cross-entropy loss to compute the classification loss. The overall training objective combines the classification loss and the previously introduced cross-space alignment loss:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{align}}, \tag{12}$$

where λ serves as a hyperparameter that regulates the weight of the cross-space alignment loss $\mathcal{L}_{\text{align}}$.

During inference, GASE operates in a fully feed-forward fashion, generating predictions directly from the learned node representations without any additional adaptation, thus ensuring efficient deployment.

5 Experiment

5.1 Datasets

We evaluate our method on six representative datasets: Cora (Sen et al., 2008), PubMed (Sen et al., 2008), Ogbn-ArXiv (Hu et al., 2020), Ogbn-Products (Hu et al., 2020), ArXiv-2023 (He et al., 2024) and ElePhoto (Yan et al., 2023). Detailed dataset descriptions can be found in the Appendix.

5.2 Baselines

We compare GASE against the following four groups of representative baselines:

1) Fully fine-tuned LM methods: BERT (Devlin et al., 2019) and DeBERTa (He et al., 2021).

2) LM+GNN methods: GLEM (Zhao et al., 2023), and UniGLM (Fang et al., 2025).

3) Frozen LLM-based methods: LLaMA3-8B (Dubey et al., 2024), LLaMA2-13B (Touvron et al., 2023), Qwen2.5-7B (Yang et al., 2024a).

4) LLM-enhanced methods: GraphBridge (Wang et al., 2024), GraphAdapter (Huang et al., 2024), OFA (Liu et al., 2024), ENGINE (Zhu et al., 2024), GAugLLM (Fang et al., 2024), LanSAGNN (Li et al., 2025) and SKETCH (Zhou et al., 2025b).

5.3 Experimental Settings

To ensure a fair comparison, all methods are evaluated under identical experimental conditions. Each baseline is reported with its best performance based on the most effective GNN variant chosen from GCN, GraphSAGE, and GAT. For LLM-based methods, both our model (GASE) and other enhanced baselines share the same LLaMA3-8B backbone. Evaluation is conducted on the node classification task using accuracy as the primary metric. All results represent the mean $\pm$ standard deviation over five independent runs with random seeds 42–46, performed on a server equipped with 4$\times$NVIDIA RTX 4090 (24GB) GPUs.

5.4 Performance Analysis

Table 1 presents a comprehensive performance comparison of GASE against four categories of baselines across six benchmark datasets. Overall, GASE consistently achieves state-of-the-art performance on all six datasets. Traditional LM-based models achieve the lowest performance due to the absence of relational modeling, while LM+GNN approaches improve upon them by introducing message passing but still rely on semantically redundant, task-agnostic language embeddings. Pure LLMs offer richer semantics but remain unable to capture structural dependencies, leading to only moderate gains. In contrast, LLM-enhanced methods advance further by combining stronger semantic representations with structural modeling.

GASE outperforms even the strongest LLM-enhanced competitors, attaining an average absolute improvement of 2.49%, and surpasses LM+GNN methods by a wider margin of 3.91% on average. On the large-scale ArXiv dataset, it delivers a 2.43% absolute gain over the best prior method. These results demonstrate that by coupling training-free structure-aware semantic extraction with principled subspace-level structural injection, GASE establishes a more effective and scalable paradigm for LLM-enhanced TAG representation learning. GASE also excels in the link prediction task. Due to space limitations, further results are provided in the Appendix.

5.5 LLM Analysis

To assess the generalizability of GASE across different LLMs, we evaluate it on three additional representative LLM backbones. As shown in Table 2, GASE consistently outperforms competitors on every backbone, confirming that our method can robustly elicit graph-aware capabilities from various frozen LLMs. Notably, while LLaMA2-13B has more parameters, it slightly underperforms the more recent LLaMA3-8B, indicating that GASE effectively inherits the superior semantic represen-

Method	Cora	PubMed	Ogbn-ArXiv	ArXiv2023	Ogbn-Products	Ele-Photo
BERT	77.67 ± 1.07	87.49 ± 0.78	70.90 ± 0.18	75.11 ± 0.04	74.59 ± 0.16	67.01 ± 0.75
DeBERTa	77.40 ± 1.01	88.41 ± 1.55	70.82 ± 0.64	75.30 ± 0.60	74.14 ± 0.40	69.56 ± 0.44
GLEM	87.20 ± 0.67	89.51 ± 1.12	74.41 ± 0.38	78.62 ± 0.76	76.42 ± 0.40	80.74 ± 0.15
UniGLM	90.07 ± 1.68	90.84 ± 1.40	75.04 ± 0.71	77.74 ± 0.83	78.12 ± 0.53	80.64 ± 0.22
LLaMA3-8B	87.41 ± 0.59	89.13 ± 1.07	75.19 ± 0.42	76.91 ± 0.81	77.81 ± 0.37	78.98 ± 0.18
LLaMA2-13B	88.24 ± 1.72	88.81 ± 1.36	74.28 ± 0.01	78.36 ± 0.82	76.35 ± 0.28	78.87 ± 0.16
Qwen2.5-7B	86.32 ± 1.82	88.83 ± 1.27	73.65 ± 0.65	77.05 ± 0.91	75.56 ± 0.61	77.71 ± 0.31
GraphBridge	88.72 ± 1.03	89.65 ± 0.44	75.32 ± 0.41	78.63 ± 0.72	78.15 ± 0.24	79.27 ± 0.61
OFA	88.99 ± 0.99	88.94 ± 1.10	76.40 ± 0.90	78.68 ± 0.42	78.41 ± 0.99	78.48 ± 0.40
GAugLLM	90.25 ± 0.20	89.76 ± 0.39	75.89 ± 0.09	79.08 ± 1.06	78.97 ± 0.58	78.01 ± 0.76
ENGINE	90.49 ± 0.18	91.66 ± 1.48	75.63 ± 0.20	79.57 ± 0.15	79.56 ± 0.12	82.59 ± 0.06
GraphAdapter	91.33 ± 0.86	91.25 ± 1.42	76.89 ± 0.37	79.49 ± 0.71	79.23 ± 0.94	80.55 ± 0.24
LanSAGNN	91.18 ± 1.56	91.44 ± 0.75	76.59 ± 0.48	79.74 ± 0.51	79.68 ± 0.42	81.72 ± 1.09
SKETCH	90.61 ± 0.83	91.56 ± 0.32	76.77 ± 1.05	79.59 ± 0.56	79.73 ± 0.58	80.53 ± 1.17
GASE	**93.46 ± 1.47**	**94.94 ± 0.83**	**79.32 ± 0.37**	**83.43 ± 0.92**	**82.64 ± 0.15**	**83.11 ± 0.79**

Table 1: Experimental results of node classification. GASE and all LLM-enhanced baselines are implemented based on LLaMA3-8B. Best results are in bold and the second-best results are underlined.

LLM	Method	Ogbn-ArXiv	Ogbn-Products
LLaMA2-13B	GASE	79.18 ± 0.87	82.36 ± 1.14
	SKETCH	76.81 ± 0.32	80.34 ± 0.69
	ENGINE	75.32 ± 1.05	79.47 ± 0.18
Qwen2.5-7B	GASE	78.84 ± 0.56	81.96 ± 0.91
	SKETCH	76.25 ± 0.12	79.40 ± 1.19
	ENGINE	75.21 ± 0.77	79.18 ± 0.45
Gemma2-9B	GASE	78.56 ± 0.95	82.05 ± 0.27
	SKETCH	76.19 ± 0.63	79.95 ± 0.08
	ENGINE	76.24 ± 1.10	80.52 ± 0.53

Table 2: Experimental results with different Large Language Model backbones.

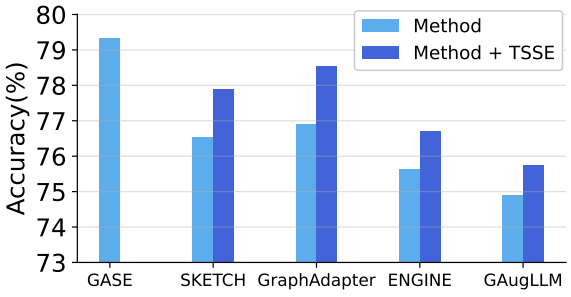


Figure 4: Plug-and-play analysis of the TSSE module.

tations of LLaMA3 and validates our semantic preservation strategy. Furthermore, GASE adapts successfully to distinct model architectures such as Qwen2.5 and Gemma2, underscoring its potential as a general-purpose framework for TAG learning.

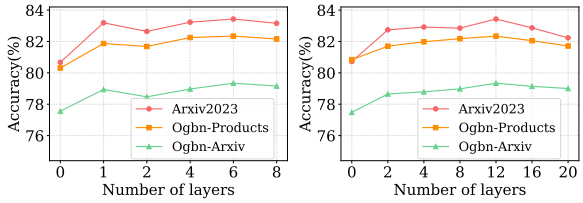
5.6 TSSE Analysis

To validate the universality of TSSE as a standalone feature enhancer, we integrate it into three existing LLM-enhanced baselines. As shown in Figure 4, adding TSSE leads to consistent and significant performance gains. In particular, GraphAdapter obtains the largest improvement, with accuracy rising by about 1.6%, while SKETCH and ENGINE also show robust gains. These results confirm that TSSE serves as an efficient, plug-and-play universal enhancer for TAG representation learning.

5.7 Ablation Study

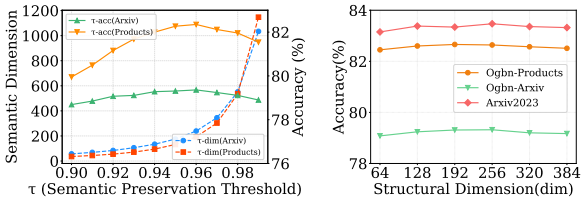
We conduct ablation studies to evaluate the contribution of each component in GASE, as summarized in Table 3. Removing the entire TSSE module leads to a significant performance drop. Ablating specific mechanisms inside TSSE, such as the Inter-layer Semantic Feedback (ISF) or Progressive Masking, also degrades results, indicating that raw LLM outputs alone are insufficient and that both mechanisms are essential for extracting high-quality neighborhood contexts.

Removing the SDSI module causes an even sharper decline, confirming that directly projecting high-dimensional semantic embeddings into a low-dimensional GNN space results in semantic loss and entangles dominant semantic directions with structural signals, thereby weakening node discriminability. In contrast, orthogonal injection via a subspace effectively fuses structure while preserving semantics. Additionally, ablating Subspace Clipping or Cross-Space Alignment also reduces performance, underscoring that appropriate subspace dimensionality and semantic constraints are critical for model robustness.



(a) Phase 1 layers L_{sem} (b) Phase 2 layers L_{str}

Figure 5: Impact of the number of layers used in Phase 1 and Phase 2 in the TSSE module.



(a) Threshold τ (b) Structural dimension d_n

Figure 6: Analysis of threshold τ and structural embedding dimension across datasets.

5.8 Hyper-parameter Analysis

Number of Layers L_{sem} and L_{str}. As shown in Figure 5(a), performance improves overall as the semantic depth L_{sem} increases, and becomes relatively stable after around $L_{\text{sem}} = 6$, indicating that shallow semantic extraction is insufficient for the ISF mechanism to fully aggregate neighborhood context. Figure 5(b) illustrates that accuracy rises with L_{str} and reaches an optimum around $L_{\text{str}} = 20$, suggesting that sufficient interaction layers are necessary for structural awareness. Beyond this point, performance degrades, which we attribute to an effect similar to over-smoothing, where excessive global aggregation in the attention mechanism may dilute the semantics of the center node.

Semantic Preservation Threshold τ and d_n. As shown in Figure 6(a), the semantic subspace dimension d_s grows slowly at first as the threshold τ increases, and then surges when $\tau > 0.96$. Accuracy peaks around $\tau = 0.96$ and subsequently drops, indicating that while a larger τ retains more semantic information, excessive dimensionality can introduce redundancy or reduce the efficiency of structural signal injection, thereby harming overall performance. Figure 6(b) shows that with d_s fixed at 256, the accuracy remains largely stable as the structural dimension d_n varies. Performance exhibits only minor fluctuations across a wide range of d_n, indicating that GASE is robust to the choice of structural embedding dimension.

	Ogbn-ArXiv	Ogbn-Products
GASE	**79.32 ± 0.37**	**82.64 ± 0.15**
w/o TSSE	76.25 ± 0.35	79.21 ± 0.89
w/o ISF	77.91 ± 1.03	80.29 ± 0.52
w/o Mask	76.73 ± 0.19	79.79 ± 0.77
w/o SDSI	76.16 ± 0.68	79.47 ± 1.12
w/o Clip	76.72 ± 0.87	80.68 ± 0.43
w/o Align	78.19 ± 0.54	81.38 ± 0.96

Table 3: Experimental results of the ablation study.

Methods	**Training Time**	**Inference Time**
GLEM$_{\text{(DeBERTa)}}$	46h 27min	51min 33s
UniGLM$_{\text{(Sentence-BERT)}}$	19h 44min	23min 16s
GraphBridge$_{\text{(LLaMA3)}}$	13h 55min	10min 37s
OFA$_{\text{(LLaMA3)}}$	10h 18min	9min 52s
GAugLLM$_{\text{(LLaMA3)}}$	8h 22min	6min 35s
GASE$_{\text{(LLaMA3) }w/o\text{ Caching}}$	8h 39min	2min 21s
GASE$_{\text{(LLaMA3)}}$	1h 43min	2min 21s

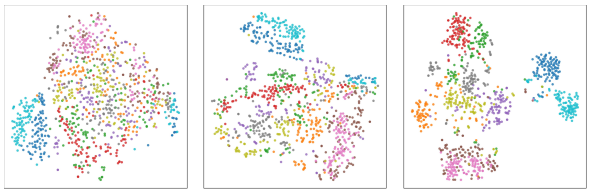
Table 4: Training and Inference Efficiency on Ogbn-ArXiv. Results were obtained on a server with a 48-core Intel Xeon CPU @ 2.10GHz and 4× NVIDIA GeForce RTX 4090 GPUs.

5.9 Efficiency Analysis

As shown in Table 4, GASE achieves substantially higher training and inference efficiency on the Ogbn-ArXiv dataset. The reported training time of GASE includes all required stages, including neighbor sampling, caching, semantic extraction, and decomposition. Under the default configuration of each method, it reduces end-to-end training time by up to 5× compared to LLM-enhanced baselines, and also achieves an order-of-magnitude improvement over LM+GNN-based methods. Caching effectively removes redundant LLM forward passes, leading to a substantial reduction in training time. Moreover, GASE exhibits the fastest inference speed, with up to 3× acceleration over GAugLLM and a significant improvement over other baselines.

5.10 Visualization

To intuitively understand the contribution of each module, we visualized the node embeddings using t-SNE. As shown in Figure 7, embeddings generated directly by the frozen LLM exhibit significant overlap between classes. Embeddings generated by the TSSE module produce tighter clusters, validating that our Structure-Aware Semantic Extraction strategy successfully refines semantic representations. The output of the full GASE framework achieves the clearest separation and high intra-class compactness, confirming its effectiveness.



(a) Llama3-8 (b) TSSE (c) GASE

Figure 7: Embedding visualization on the ArXiv dataset using the same randomly sampled nodes.

6 Conclusion

In this work, we present GASE, a framework that adapts frozen LLMs into graph-aware semantic encoders for TAG learning. By introducing a training-free structure-aware semantic extraction module and a subspace-based structural injection strategy, GASE effectively integrates graph structure while preserving the semantic geometry. Extensive experiments show that GASE achieves state-of-the-art performance with significantly improved efficiency, enabling an efficient and scalable approach to leveraging frozen LLMs for large-scale graph learning.

Limitations

This work focuses on node-centric semantics and does not explicitly incorporate edge attributes, which can be critical for certain graph tasks, such as molecule property prediction, where each edge may possess distinct properties. While most TAG datasets rely primarily on node information, incorporating edge features may further broaden the applicability. In future work, we will expand our framework to integrate edge attributes. Moreover, GASE does not explicitly support online graph learning. When graph structures or node texts change over time, cached TSSE representations may become outdated. Extending the framework with incremental semantic updating mechanism remains an important direction for future work.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and 1 others. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. 2023. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. 2025. Uniglm: Training one unified language model for text-attributed graphs embedding. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 973–981.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Guoqing Hu, An Zhang, Shuo Liu, Zhibo Cai, Xun Yang, and Xiang Wang. 2025. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1614–1623.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024. Let's ask GNN: Empowering large language model for graph in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can gnn be good adapter for llms? In *Proceedings of the ACM Web Conference 2024*, pages 893–904.

Zhaoxing Li, Xiaoming Zhang, Haifeng Zhang, and Chengxiang Liu. 2025. Refining interactions: Enhancing anisotropy in graph neural networks with language semantics. *arXiv preprint arXiv:2504.01429*.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2025. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12183–12191.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Mag.*, 29(3):93–106.

Matthieu Tehenan. 2025. Semantic geometry of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11993–12004.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413.

Yaoke Wang, Yun Zhu, Wenqiao Zhang, Yueting Zhuang, Liyunfei Liyunfei, and Siliang Tang. 2024. Bridging local details and global context in text-attributed graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 14830–14841.

Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. 2025. Can llms convert graphs to text-attributed graphs? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 1412–1432. Association for Computational Linguistics.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10370–10388.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, and 1 others. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, and et al Haoran Wei. 2024a. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Haoran Yang, Xiangyu Zhao, Sirui Huang, Qing Li, and Guandong Xu. 2024b. Latex-gcl: Large language models (llms)-based data augmentation for text-attributed graph contrastive learning. *arXiv preprint arXiv:2409.01145*.

Liangwei Yang, Jing Ma, Jianguo Zhang, Zhiwei Liu, Jielin Qiu, Shirley Kokane, Shiyu Wang, Haolin Chen, Rithesh Murthy, Ming Zhu, and 1 others. 2025a. Geognn: Quantifying and mitigating semantic drift in text-attributed graphs. *arXiv preprint arXiv:2511.09042*.

Mengxue Yang, Chun Yang, Jiaqi Zhu, Jiafan Li, Jingqi Zhang, Yuyang Li, and Ying Li. 2025b. SLiNT: Structure-aware language model with injection and contrastive training for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. 2025. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13087–13095.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. 2024. Text-attributed graph representation learning: Methods, applications, and chal-

lenges. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1298–1301.

Zihao Zhang, Xunkai Li, Rong-Hua Li, Bing Zhou, Zhenjun Li, and Guoren Wang. 2025. Toward general and robust llm-enhanced text-attributed graph learning. *CoRR*, abs/2504.02343.

Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Chenhui Zhang, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. 2024. Pre-training and prompting for few-shot node classification on text-attributed graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4467–4478.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Chuang Zhou, Jiahe Du, Huachi Zhou, Hao Chen, Feiran Huang, and Xiao Huang. 2025a. Text-attributed graph learning with coupled augmentations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10865–10876.

Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. 2025b. Taming language models for text-attributed graph learning with decoupled aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3463–3474.

Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient tuning and inference for large language models on textual graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 5734–5742.

A Appendix

A.1 Theoretical Analysis

Figure 3 shows the variance spectrum of the language embedding matrix $\mathbf{E}$, where larger spectral values correspond to directions with higher embedding variance. Prior empirical studies have shown that semantic information in language embeddings is largely concentrated in a small number of high-variance directions (Tehenan, 2025). Motivated by this observation, we decompose the embedding space into a semantic-rich subspace and a complementary residual subspace.

We analyze the centered embedding matrix $\mathbf{E}-\mu$ through its variance spectrum. The semantic-rich subspace is defined by the top d_s directions associated with the largest spectral values, which preserve most of the semantic information. Restricting embeddings to this subspace yields an optimal rank-d_s approximation under the Frobenius norm. The remaining low-variance directions therefore provide a suitable, low-interference space for injecting structural signals.

Eckart-Young Theorem

Theorem Statement

Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with rank r and its singular value decomposition (SVD) $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, then for any integer $k \leq r$, the optimal rank-k approximation of $\mathbf{A}$ under Frobenius norm is:

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top \tag{13}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ consist of the first k columns of $\mathbf{U}$ and $\mathbf{V}$ respectively, and $\mathbf{\Sigma}_k = \text{diag}(\sigma_1, \ldots, \sigma_k)$. The approximation error is:

$$\min_{\text{rank}(\mathbf{B})\leq k} \|\mathbf{A}-\mathbf{B}\|_F = \|\mathbf{A}-\mathbf{A}_k\|_F = \sqrt{\sum_{j=k+1}^{r} \sigma_j^2} \tag{14}$$

$$\min_{\text{rank}(\mathbf{B})\leq k} \|\mathbf{A}-\mathbf{B}\|_2 = \|\mathbf{A}-\mathbf{A}_k\|_2 = \sigma_{k+1} \tag{15}$$

Connection to GASE

Within the SDSI module, the dimension d_s is determined by an energy retention criterion:

$$d_s = \min\left\{ k \;\middle|\; \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{j=1}^{r}\sigma_j^2} \geq \tau \right\}, \tag{16}$$

where $\tau = 0.95$ controls the proportion of preserved embedding variance.

For the LLM embedding matrix $\mathbf{E} \in \mathbb{R}^{N\times d_l}$, the Eckart–Young–Mirsky theorem guarantees that the best rank-d_s approximation $\mathbf{E}'$ under the Frobenius norm is obtained by truncating the SVD at d_s, with reconstruction error:

$$\min_{\text{rank}(\mathbf{E}')\leq d_s} \|\mathbf{E}-\mathbf{E}'\|_F^2 = \sum_{i=d_s+1}^{r} \sigma_i^2. \tag{17}$$

The residual subspace spanned by $\{\mathbf{v}_{d_s+1}, \ldots, \mathbf{v}_r\}$ corresponds to low-variance directions of the embedding distribution. Empirically, these directions exhibit weaker semantic signals, which motivates their use for structural signal injection while mitigating interference with dominant semantic components.

Dataset	Avg.#Nodes	Avg.#Edges	#Graphs	#C	Domain	Task	Metric
Cora	2,708	5,429	1	7	Academic	Node	Accuracy
Pubmed	19,717	44,338	1	3	Academic	Node, Link	Accuracy, AUC
Ogbn-ArXiv	169,343	1,166,243	1	40	Academic	Node, Link	Accuracy, AUC
ArXiv2023	46,198	78,548	1	40	Academic	Node, Link	Accuracy, AUC
Ogbn-Products	54,025	74,420	1	47	E-commerce	Node, Link	Accuracy, AUC
Ele-Photo	48,362	500,928	1	12	E-commerce	Node, Link	Accuracy, AUC

Table 5: Datasets

Method	PubMed	Ogbn-ArXiv	ArXiv2023	Ogbn-Products	Ele-Photo
GLEM	88.04 ± 0.87	82.87 ± 0.23	91.14 ± 0.56	87.96 ± 0.12	90.34 ± 0.45
UniGLM	88.69 ± 0.34	84.95 ± 0.98	92.21 ± 0.37	89.31 ± 0.67	91.11 ± 0.29
Llama3-8B	87.69 ± 1.12	83.95 ± 0.45	91.21 ± 0.73	86.96 ± 1.08	89.34 ± 0.91
Llama2-13B	87.04 ± 0.62	82.87 ± 1.50	90.10 ± 0.89	88.31 ± 0.84	90.35 ± 1.19
GraphBridge	89.29 ± 0.77	85.91 ± 0.31	92.32 ± 0.42	90.20 ± 0.53	90.54 ± 0.68
OFA	90.37 ± 1.04	84.97 ± 0.89	92.89 ± 0.36	89.14 ± 1.33	92.18 ± 0.47
ENGINE	88.71 ± 0.18	85.38 ± 1.15	92.48 ± 0.71	88.87 ± 0.39	92.04 ± 1.27
LanSAGNN	90.13 ± 0.95	84.46 ± 0.63	92.74 ± 0.09	90.72 ± 0.22	90.89 ± 0.76
SKETCH	89.34 ± 0.51	85.21 ± 0.30	92.11 ± 0.44	90.29 ± 0.97	91.20 ± 0.13
GASE	$\mathbf{93.52 \pm 0.81}$	$\mathbf{89.14 \pm 0.57}$	$\mathbf{96.83 \pm 0.21}$	$\mathbf{93.87 \pm 0.65}$	$\mathbf{95.46 \pm 0.83}$

Table 6: Results of link prediction. GASE and all LLM-enhanced baselines are implemented based on Llama3-8B.

A.2 Datasets

In this section, we present a comprehensive overview of the datasets utilized in this paper. The details of these datasets are as follows: **Cora** (Sen et al., 2008) dataset consists of 2,708 scientific publications categorized into seven classes: Case-based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, and Theory. Each paper in the citation network cites or is cited by at least one other paper, resulting in a total of 5,429 edges.

Pubmed (Sen et al., 2008) dataset includes 19,716 nodes, 44,338 edges, and 59,381 texts, including both titles and abstracts. The included articles are about diabetes and the standard node categories are from the Pubmed database: type-1 diabetes, type-2 diabetes, or experimental evidence. Each category addresses distinct facets of diabetes research, contributing to the understanding and treatment of this multifaceted disease.

Ogbn-ArXiv (Hu et al., 2020) dataset represents a directed graph showcasing the citation network among computer science arXiv papers. Each node signifies an arXiv paper, with directed edges denoting citations. The goal is to categorize papers into one of 40 subject areas such as cs.AI, and cs.OS,

Ogbn-Products (Hu et al., 2020) dataset includes 54k nodes and 74k edges. Each node in this dataset represents products sold on Amazon, and edges between two products indicate that the products are purchased together. The task involves predicting the category of a product in a multi-class classification setup, using the 47 top-level categories as target labels.

ArXiv2023 (He et al., 2024) dataset, is a directed graph representing the citation network of computer science arXiv papers published in 2023 or later. Similar to Ogbn-ArXiv, it features nodes representing arXiv papers and directed edges for citations. The goal is to classify each paper into one of 40 subject areas such as cs.AI, cs.LG, and cs.OS, with classifications provided by the authors and arXiv moderators.

Ele-Photo (Yan et al., 2023) dataset, derived from the AmazonElectronics dataset, consists of nodes representing electronics products. Each node is labeled according to a three-level classification of electronics products. The text attribute for each node is the user review with the most votes, or a randomly selected review if no highly-voted reviews are available. The task is to classify these products into 12 categories.

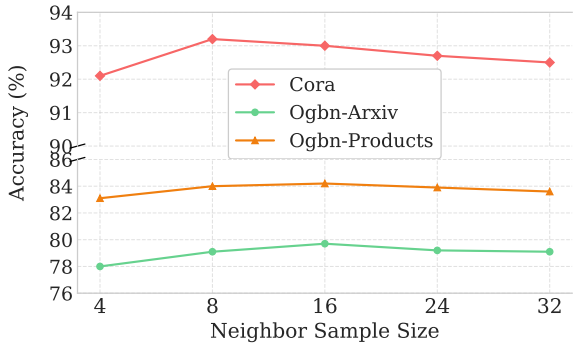


Figure 8: Impact of neighbor sample size K across three datasets with varying scales.

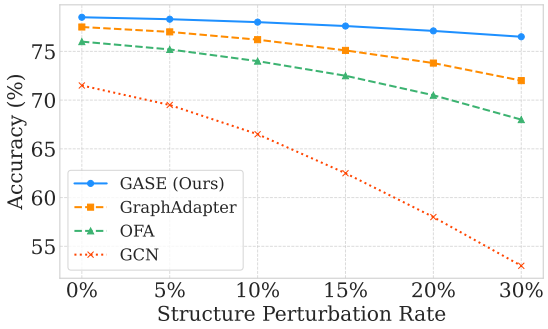


Figure 9: Robustness analysis against structural perturbation on the Ogbn-ArXiv dataset.

A.3 Link Prediction Evaluation

Our method is not limited to node classification tasks, it can also be applied to edge-level or graph-level tasks. In this section, we conduct experiments on link prediction tasks. We split existing edges into train:val:test=0.85:0.05:0.1 for all datasets. The reported results include the mean AUC with standard deviation across 5 different random seeds. We use traditional GNN methods, LM+GNN methods and LLM enhanced methods as baselines and present the results of our method. According to Table 6, GASE outperforms all baselines by a considerable margin on the five datasets. In particular, GASE exhibits more than a 3% absolute improvement on the ArXiv2023 dataset. These statistics underscore the effectiveness of our method in edge-level tasks.

A.4 Neighbor Sampling Numbers Analysis

To verify the generalization capability of GASE regarding neighborhood aggregation, we evaluated the impact of neighbor sample size K on three distinct datasets with different densities and scales: Cora, Ogbn-ArXiv, and Ogbn-Products.As shown in Figure 8, GASE exhibits a consistent robustness pattern across all datasets. Performance improves rapidly as K increases from 4 to 8, indicating effective context incorporation. Crucially, when K exceeds 16, GASE avoids the performance degradation often observed in concatenation-based baselines due to noise accumulation. This confirms that our TSSE module can adaptively compress and filter neighborhood information.

A.5 Robustness to Structural Noise

To verify the robustness of GASE against structural noise, we conducted a perturbation experiment by randomly adding edges to the graph structure, with ratios ranging from 0% to 30%. As illustrated in Figure 9, traditional GCNs suffer a catastrophic performance drop as the structure deteriorates, validating their heavy reliance on topological quality. Compared with other LLM-enhanced methods, GASE exhibits a remarkably flat degradation curve, which is attributed to our SDSI strategy. By isolating the semantic-rich subspace derived from the frozen LLM, GASE effectively establishes a semantic lower bound, ensuring that the model retains strong discriminative power even when structural signals are unreliable.

A.6 Ablation on Progressive Masking in TSSE

To verify the necessity of our progressive masking strategy, we conducted an ablation study by removing specific phases from the TSSE module, as shown in Table 7. Specifically, (1) *w/o Phase 1 & 2:* Standard causal attention yields the lowest performance, confirming that applying raw attention without compression or structural constraints leads to severe attention dilution. (2) *w/o Phase 2:* No structural guide performs poorly, indicating that even with compressed semantics, the model fails to effectively distinguish neighbor importance without the star-topology constraint. (3) *w/o Phase 1:* Removing compression incurs performance losses due to noise accumulation, as the model is forced to attend to raw, lengthy neighbor sequences. (4) *w/o Phase 3:* No global integration comes closest to the best performance but still lags behind. In summary, TSSE achieves the best performance, validating that our progressive masking is an effective strategy for graph-aware semantic extraction.

A.7 Impact of Projection Strategies

To validate the superiority of our SDSI strategy over standard dimensionality reduction techniques,

Method	PubMed	Ogbn-ArXiv	ArXiv2023	Ogbn-Products	Ele-Photo
Ours (Progressive)	**94.94 ± 0.83**	**79.32 ± 0.37**	**83.43 ± 0.92**	**82.64 ± 0.15**	**83.11 ± 0.79**
-*w/o* phase 3	94.05 ± 0.92	78.63 ± 0.34	82.74 ± 0.17	81.79 ± 0.72	82.26 ± 0.56
-*w/o* phase 2	91.27 ± 1.35	77.48 ± 0.52	80.72 ± 0.89	80.84 ± 0.41	81.36 ± 0.23
-*w/o* phase 1	91.38 ± 0.47	77.56 ± 0.28	80.67 ± 0.63	80.31 ± 0.15	81.59 ± 1.39
-*w/o* phase 2&1	90.17 ± 1.68	76.73 ± 0.19	79.68 ± 1.04	79.79 ± 0.77	79.62 ± 0.98

Table 7: Performance comparison of different masking strategies on Ogbn-ArXiv dataset.

Method	PubMed	Ogbn-ArXiv	ArXiv2023	Ogbn-Products	Ele-Photo
SDSI	**94.94 ± 0.83**	**79.32 ± 0.37**	**83.43 ± 0.92**	**82.64 ± 0.15**	**83.11 ± 0.79**
MLP	91.21 ± 1.12	76.35 ± 0.76	79.54 ± 0.32	79.89 ± 1.45	80.36 ± 0.91
PCA	89.18 ± 0.83	75.67 ± 0.61	79.71 ± 1.08	78.85 ± 0.37	80.29 ± 1.07

Table 8: Comparison of SDSI with MLP and PCA Projection Modules.

Strategy	Ogbn-ArXiv	Ogbn-Products
ISF	**79.32 ± 0.37**	**82.64 ± 0.15**
<EOS>	78.25 ± 0.37	81.29 ± 0.82
Mean Pooling	78.41 ± 0.54	81.16 ± 0.19

Table 9: Comparison of different neighbor compression strategies.

we compared GASE with two variants. As shown in Table 8, the PCA variant gets the worst performance as it is non-trainable linear compression and may discard low-variance dimensions that are semantically sparse but structurally significant. The MLP variant also get poor performance. We attribute this to: MLP projects high-dimensional LLM embeddings into a low-dimensional space indiscriminately. This process inevitably mixes dominant semantic features with injected structural signals, causing the graph structure to act as noise that distorts the pretrained semantic geometry. In contrast, SDSI preserves the LLM semantics while effectively incorporating graph structures.

A.8 Impact of Neighbor Compression Strategies

To verify the effectiveness of the Inter-layer Semantic Feedback (ISF) mechanism in Phase 1 in the TSSE module , we compared it against two alternative neighbor representation strategies: (1) using the standard EOS token, (2) Mean Pooling over all tokens. As shown in Table 9, the ISF strategy achieves the highest accuracy. The inferior performance of Mean Pooling and EOS Only indicates that simple aggregation methods fail to capture the complex semantic dependencies within neighbor texts. Standard pooling blurs distinct features, while the un-tuned EOS token lacks the capability to aggregate global context effectively.

A.9 Implementation Details

For neighbor sampling, we sample a fixed-size neighbor set of K nodes for each center node v by prioritizing local structure: we first uniformly sample without replacement from its 1-hop neighbors, then from its shortest-path 2-hop neighbors (excluding v and 1-hop nodes). If fewer than K neighbors are obtained, we run random walks starting from v to collect additional unique nodes until reaching K, and cache all sampled lists offline with a fixed random seed for reproducibility. To ensure a fair comparison, we use exactly the same neighbor sampling strategy for all baselines. We use the decoder-only LLM Llama3-8B as the semantic backbone and fully freeze all LLM parameters; consequently, no gradients are backpropagated through the LLM in TSSE stage. Embeddings generated by TSSE are precomputed and cached prior to SDSI. For SDSI, we perform a one-time truncated SVD on the cached node embedding matrix E to obtain the semantic-rich subspace, which is reused throughout training.