AUDITING PRIVACY RISKS IN LLM-ENHANCED GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have recently advanced graph neural networks (GNNs) by enriching node representations with semantic information, giving rise to *LLM-enhanced GNNs* that achieve substantial performance gains. However, how such semantic enhancement affects privacy risks remains largely underexplored. To bridge this gap, we systematically audit the privacy risks of LLM-enhanced GNNs through a unified framework consisting of five stages: ① dataset preparation, ② victim model training, ③ privacy attack, ④ risk assessment, and ⑤ defense analysis. Specifically, our evaluation spans ten text-attributed graph datasets across diverse domains, six privacy attacks, 42 LLM-enhanced GNN configurations, and three more recent language-model backbones. Extensive experiments show that, despite their utility improvements, LLM-enhanced GNNs consistently exhibit greater empirical privacy vulnerability than shallow text representation baselines under the evaluated attacks across diverse models and datasets. Further analysis shows that LLM-enhanced representations exhibit more distinguishable link-, label-, and membership-related signals in the embedding space, making them more exploitable by inference attacks. Finally, we evaluate representative defenses and examine their effectiveness in mitigating these privacy risks. Overall, this work provides a systematic audit of privacy risks in LLM-enhanced GNNs and offers insights for developing more secure and trustworthy graph learning systems.

Dataset: Anonymous link. Code: Anonymous link.

1 INTRODUCTION

In recent years, graph neural networks (GNNs) (Wu et al., 2020b; Kipf & Welling, 2017) have achieved remarkable success in graph representation learning and have become the dominant paradigm for modeling graph-structured data. In many real-world applications, graphs are not only characterized by their topology but also associated with rich textual attributes, such as user profiles, paper abstracts, or semantic descriptions of entities, forming so-called text-attributed graphs (Yan et al., 2023; Yao et al., 2019). Traditional approaches typically rely on shallow text encoding techniques, such as Bag-of-Words (BoW) (Harris, 1954), Word2Vec (Church, 2017), or skip-gram models (Mikolov et al., 2013), to obtain initial node representations, which are then integrated into GNN-based message passing frameworks. However, these methods are limited in their ability to capture rich contextual and semantic information in node texts. Consequently, effectively modeling textual semantics while preserving structural dependencies remains a key challenge in graph representation learning.

Recently, large language models (LLMs) have demonstrated strong capabilities in natural language understanding (Kuang et al., 2025; Yi et al., 2025), contextual reasoning (Xiong et al., 2025; Chi et al., 2024), and instruction following (Lou et al., 2024; Qin et al., 2024), and have been increasingly introduced into graph learning to enhance the semantic modeling of textual attributes. Compared with traditional text encoders, LLMs can generate more expressive and context-aware representations by capturing deep semantic relationships beyond surface-level word co-occurrence patterns. This has led to the emergence of *LLM-enhanced GNNs* (He et al., 2024; Chen et al., 2024; Zhu et al., 2024; Chien et al., 2022; Zhang et al., 2025b), a new paradigm that integrates LLMs into graph learning to enrich node representations. As illustrated in Fig. 1(II), existing approaches can be broadly categorized into *explanation-based* and *embedding-based* approaches. These approaches typically utilize LLMs or smaller pretrained language models (LMs) to perform semantic augmentation,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

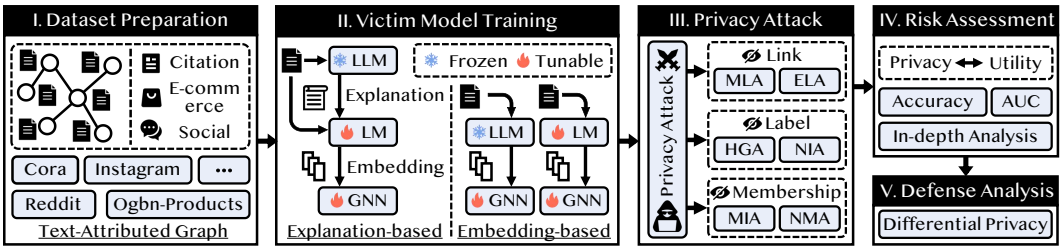


Figure 1: **Overview of the proposed privacy auditing framework.** It consists of five stages: dataset preparation, victim model training, privacy attack, risk assessment, and defense analysis.

such as contextual enrichment or representation refinement of node texts, and seamlessly integrate the resulting embeddings into GNN pipelines. By jointly modeling structural dependencies and enriched semantic representations, LLM-enhanced GNNs have achieved substantial performance improvements across various graph learning tasks, such as node classification (He et al., 2024).

Despite their strong empirical performance, the privacy risks of LLM-enhanced GNNs remain largely underexplored. Existing studies (Meng et al., 2023; Zhang et al., 2022; He et al., 2021b) have shown that GNNs are highly vulnerable to privacy attacks, where adversaries can infer sensitive information from model outputs, even when only limited access is available. In LLM-enhanced GNNs, such risks may be further exacerbated, as node representations integrate structural dependencies with rich semantic features encoded by LLMs or LMs, potentially increasing the information content exposed through model outputs. This paradigm shift may introduce new and more powerful avenues for privacy leakage beyond those observed in conventional GNNs. Therefore, how privacy vulnerabilities evolve under LLM-enhanced architectures remains poorly understood, calling for a systematic audit of the privacy risks associated with this emerging graph learning paradigm.

To fill this gap, we systematically audit the privacy risks of LLM-enhanced GNNs. As illustrated in Fig. 1, we develop a unified privacy auditing framework encompassing five stages, including ① dataset preparation, ② victim model training, ③ privacy attack, ④ risk assessment, and ⑤ defense analysis, enabling an end-to-end characterization of privacy vulnerabilities and mitigation strategies.

Specifically, our evaluation spans ten real-world text-attributed graph datasets across diverse domains, with six used in the main benchmark and four for additional cross-domain validation. We evaluate 42 LLM-enhanced GNN configurations across multiple feature enhancers and widely used GNN backbones, together with three more recent language-model backbones, covering diverse model architectures and representations. We then consider six representative privacy attacks targeting three fundamental objectives, namely *link*, *label*, and *membership* inference. These attack objectives have been extensively studied in prior work (Olatunji et al., 2021; Meng et al., 2023; Zhang et al., 2022; He et al., 2021b; Guan et al., 2025; Zhang et al., 2024) on traditional GNNs and are widely recognized as the most prevalent forms of privacy leakage, as they respectively capture the exposure of graph structure, node semantics, and training participation. In the risk assessment stage, we conduct a quantitative evaluation of privacy risks and further investigate how semantic enhancement affects model vulnerability. The results show that LLM-enhanced representations exhibit stronger structural-, label-, and membership-related signals, providing a plausible explanation for their higher empirical privacy vulnerability. Finally, we evaluate representative differential privacy (DP) (Sajadmanesh & Gatica-Perez, 2021; Lin et al., 2022; Zhu et al., 2023; He et al., 2025; Dwork, 2008) defenses covering both input-level perturbation and private GNN training, and characterize their effectiveness and utility costs under different protection settings. Our contributions can be summarized as follows:

- ◇ **Important But Underexplored Problem.** We identify and systematically investigate the important yet underexplored privacy risks in LLM-enhanced GNNs, which have received limited attention in existing graph learning and privacy research.
- ◇ **Systematic Auditing Framework.** We develop a unified five-stage privacy auditing framework to systematically evaluate privacy risks in LLM-enhanced GNNs, covering real-world datasets, diverse privacy attacks, in-depth risk analysis, and representative defenses, thereby enabling a holistic understanding of privacy vulnerabilities and mitigation effectiveness across settings.
- ◇ **Comprehensive Empirical Evaluation.** Our evaluation spans ten text-attributed graph datasets, six privacy attacks, 42 LLM-enhanced GNN configurations, and three more recent language-model backbones, together with extensive analyses of representative privacy defenses.

2 PRELIMINARIES

In this section, we first present the problem definition (▷ Sec. 2.1), and then introduce the necessary background on graph neural networks (▷ Sec. 2.2) and LLM-enhanced GNNs (▷ Sec. 2.3).

2.1 PROBLEM DEFINITION

Consider a text-attributed graph (TAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathbf{A})$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ denotes the set of nodes, $\mathcal{E}$ represents the set of edges, $\mathcal{T}=\{t_v\}_{v\in\mathcal{V}}$ contains the textual attributes of the nodes, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix with $\mathbf{A}_{ij} \in \{0, 1\}$ indicating the presence of an edge between nodes v_i and v_j. TAGs seamlessly integrate structured topological information with unstructured textual semantics, providing a versatile representation for a wide range of real-world networks, including social networks, citation networks, and e-commerce networks. Consistent with prior work (He et al., 2024; Chen et al., 2024; Zhu et al., 2024; Chien et al., 2022; Zhang et al., 2025b), we focus on the node classification task. The node set $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_U$ consists of labeled nodes $\mathcal{V}_L$ and unlabeled nodes $\mathcal{V}_U$, where each labeled node $v \in \mathcal{V}_L$ is associated with a class label $y_v \in \mathcal{Y} = \{1, \ldots, C\}$. The objective is to learn a mapping $f_\theta : (\mathbf{A}, \mathcal{T}) \to \mathcal{Y}$ that predicts labels for unlabeled nodes by jointly exploiting graph structure and textual node attributes. Building on this formulation, we investigate how an adversary can exploit the outputs of LLM-enhanced GNNs, together with their prior knowledge and capabilities, to infer sensitive information about target nodes, including link existence, node labels, and training membership (see Sec. 3 for the threat model).

2.2 GRAPH NEURAL NETWORKS

GNNs are a class of deep learning models designed for graph-structured data, whose core idea is to learn node representations (*i.e.*, embeddings) by aggregating information from neighboring nodes (Wu et al., 2020b). Taking advantage of this message passing mechanism, GNNs have achieved strong performance across a wide range of real-world applications, including social network analysis (Kipf & Welling, 2017), knowledge graph completion (Liu et al., 2021), fraud detection (Yang et al., 2025), and recommender systems (Wu et al., 2022b). Representative architectures include GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), and SAGE (Hamilton et al., 2017). Formally, let $\mathcal{N}(v)$ denote the set of neighbors of node v. At the k-th layer, node representations are computed via a two-step procedure consisting of neighborhood aggregation and representation update:

$$\mathbf{h}^k_{\mathcal{N}(v)} = \textsc{Aggregate}_k(\{\mathbf{h}^{k-1}_u \mid u \in \mathcal{N}(v)\}), \quad \mathbf{h}^k_v = \textsc{Update}_k(\mathbf{h}^{(k-1)}_v, \mathbf{h}^k_{\mathcal{N}(v)}), \tag{1}$$

where $\mathbf{h}^{k-1}_u$ represents the representation of neighbor u at layer $k-1$. The operator $\textsc{Aggregate}_k(\cdot)$ performs permutation-invariant aggregation over neighbors, such as mean, sum, or max pooling, while $\textsc{Update}_k(\cdot)$ denotes a learnable non-linear transformation, typically parameterized by a neural network. At the input layer, $\mathbf{h}^0_v$ is initialized with the original feature vector $\mathbf{x}_v$ of node v.

2.3 LLM-ENHANCED GNNS

With the rapid advancement of LLMs, recent studies (He et al., 2024; Chen et al., 2024; Zhu et al., 2024; Chien et al., 2022) have explored integrating them with GNNs, giving rise to *LLM-enhanced GNNs*[1]. These approaches leverage LLMs to encode textual node attributes, introducing rich semantic information that complements the limited expressiveness of traditional GNNs on unstructured text. As a result, they can significantly improve node representations and downstream task performance. Depending on whether additional textual information is generated by LLMs, as illustrated in Fig. 1(II), existing methods can be broadly categorized into two groups:

- ◇ **Explanation-based methods** exploit the reasoning and knowledge capabilities of LLMs to derive high-level semantic information (*e.g.*, explanations) from raw textual attributes, which is then encoded into node embeddings by an LM (*e.g.*, DeBERTa-base (He et al., 2021a)). Formally, this process can be written as $\mathbf{x}_v = f_{\text{LM}}(f_{\text{LLM}}(p, t_v), t_v)$, where f_{LLM} generates augmented text conditioned on a prompt p, and f_{LM} encodes both original and augmented text into embeddings.
- ◇ **Embedding-based methods** do not explicitly generate additional text but instead directly encode textual attributes into embeddings. These methods can be further divided into two variants: ① directly using LLMs to obtain embeddings, *i.e.*, $\mathbf{x}_v = f_{\text{LLM}}(t_v)$, and ② fine-tuning conventional

[1]Following prior work (Li et al., 2024; He et al., 2024; Chen et al., 2024; Ma et al., 2026; Zhang et al., 2025b; Yang et al., 2025), we use *LLM-enhanced GNNs* as a broad term encompassing GNNs augmented by either large language models (LLMs) or pretrained language models (LMs).

LMs to produce task-specific embeddings, *i.e.*, $x_v = f_{\text{LM}}(t_v)$. Such approaches typically rely on embedding-accessible LLMs or fine-tunable pretrained models to adapt to downstream tasks.

3 Threat Model

We characterize the threat model in terms of the *knowledge*, *capability*, and *objective* of the adversary.

Adversary's Knowledge and Capability. We consider an adversary with *black-box access* to the target GNN model. Specifically, the adversary has no access to the model's internal parameters or architecture and can only obtain node posteriors through query access. This setting is consistent with prior attack literature (Meng et al., 2023; Zhang et al., 2022; He et al., 2021b; Fu et al., 2025) and aligns with real-world deployments of *machine-learning-as-a-service* (MLaaS) platforms, where models are exposed via query interfaces, as exemplified by Google Cloud's Vertex AI (Google, 2021), Meta Platforms' ParlAI (Parlai, 2017), and IBM's InfoSphere Virtual Data Pipeline (IBM, 2018). Beyond query access, the adversary may possess auxiliary prior knowledge $\mathcal{K}$ and additional capabilities to facilitate the attack. As illustrated in Fig. 2, the adversary interacts with the model to collect posterior predictions, which are then exploited to infer sensitive information about the underlying graph data.

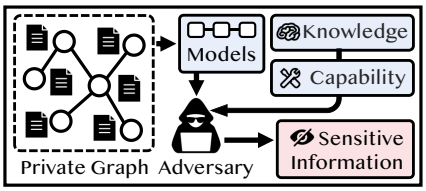


Figure 2: Adversary–model interaction illustration under black-box access.

Adversary's Objective. We focus on three fundamental privacy risks in graph learning, namely *link*, *label*, and *membership* inference. These attack objectives are widely studied in prior work (Olatunji et al., 2021; Guan et al., 2025; Meng et al., 2023; Zhang et al., 2022; Wu et al., 2022a; He et al., 2021b; Zhang et al., 2024) on traditional GNNs and are considered representative forms of privacy leakage, as they respectively target graph structure, node semantics, and training participation. By adopting this taxonomy, our evaluation remains consistent with established attack paradigms while enabling systematic assessment of privacy risks in LLM-enhanced GNNs. Formally, given a target node v and its model posterior $\mathcal{R}_v$, the adversary's objectives are defined as follows:

- **Link Inference.** The adversary attempts to infer whether a link exists between node v and another node u. This reveals structural relationships in the graph, which may correspond to sensitive interactions (*e.g.*, social ties). The objective is: $\hat{e}_{vu} = \arg\max_{e\in\{0,1\}} \Pr(e_{vu} = e \mid \mathcal{R}_v, \mathcal{R}_u)$.
- **Label Inference.** The adversary aims to infer the true class label y_v of node v. The label y_v can expose sensitive information about node v, *e.g.*, personalized recommendations that reveal their private interests. The formal objective is defined as: $\hat{y}_v = \arg\max_{c\in\mathcal{Y}} \Pr(y_v = c \mid \mathcal{R}_v)$.
- **Membership Inference.** The adversary aims to determine whether node v was included in the training set of the target model. Such information can reveal participation in sensitive datasets (*e.g.*, medical or user-specific records). The objective is: $\hat{m}_v = \arg\max_{m\in\{0,1\}} \Pr(m_v = m \mid \mathcal{R}_v)$, where $m_v \in \{0, 1\}$ indicates whether v is a member of the training data.

4 Privacy Auditing Framework

In this section, we present the five stages of our privacy auditing framework in sequence: *dataset preparation* (▷ Sec. 4.1), *victim model training* (▷ Sec. 4.2), *privacy attack* (▷ Sec. 4.3), *risk assessment* (▷ Sec. 4.4), and *defense analysis* (▷ Sec. 4.5). Fig. 1 illustrates the overall workflow.

4.1 Dataset Preparation

We consider six TAG datasets in the main evaluation, with statistics summarized in Table 1. They span three real-world scenarios: social networks, citation networks, and e-commerce platforms. The social networks include *Instagram* (Huang et al., 2024) and *Reddit* (Huang et al., 2024), where nodes represent users and textual attributes correspond to user-generated content or profiles. The citation networks include *Cora* (McCallum et al., 2000), *CiteSeer* (Yang et al., 2016), and *Tape-Arxiv23* (subset) (He et al., 2024), where nodes denote papers and

Table 1: Statistics of datasets.

Dataset	#Nodes	#Edges	#Classes	Domain
Cora	2,708	5,429	7	Citation
CiteSeer	3,327	4,552	6	Citation
Ogbn-Products	12,011	21,987	47	E-commerce
Tape-Arxiv23	13,167	23,735	40	Citation
Instagram	11,339	144,010	2	Social
Reddit	33,434	198,448	2	Social

textual attributes are derived from titles or abstracts. For e-commerce, we adopt *Ogbn-Products* (subset) (He et al., 2024), where nodes correspond to products with textual descriptions or reviews. These datasets provide diverse structural and textual characteristics for cross-domain privacy evaluation. Further details are provided in App. A.1, with four additional datasets evaluated in App. B.7.

4.2 Victim Model Training

In this stage, we consider 42 victim model configurations for training. Each configuration consists of two key components: an LM/LLM-based feature enhancer and a GNN backbone. To systematically cover different integration paradigms, we select a representative set of feature enhancement methods:

- **Explanation-based methods.** We consider *TAPE* (He et al., 2024) and *KEA* (Chen et al., 2024), two representative approaches that leverage LLMs (*e.g.*, GPT-3.5-Turbo) to generate complementary semantic information, such as explanations or keyphrases. The generated text is then jointly encoded with the original textual attributes using LMs (*e.g.*, DeBERTa-base (He et al., 2021a) or E5-Large (Wang et al., 2022)) to produce node embeddings.
- **Embedding-based methods.** As described in Sec. 2.3, these methods follow two implementation paradigms. The first directly leverages LLMs to generate text embeddings, for which we consider LLaMA-2-7b-hf (*LLaMA*) (Touvron et al., 2023) and Linq-Embed-Mistral (*Linq*) (Choi et al., 2024). The second fine-tunes the LMs to obtain task-specific embeddings; in this category, we include *SimTeg* (Duan et al., 2023) and *E5-Large* (Wang et al., 2022).

Based on these feature enhancers, we further combine them with seven GNN backbones: *GCN* (Kipf & Welling, 2017), *SAGE* (Hamilton et al., 2017), *GAT* (Velickovic et al., 2018), *GIN* (Xu et al., 2019), *APPNP* (Klicpera et al., 2019), *SGC* (Wu et al., 2019), and *SSGC* (Zhu & Koniusz, 2021). In total, this results in 42 victim model configurations. Rather than pursuing more advanced architectures, we focus on widely adopted and well-established combinations to ensure reproducibility and provide representative coverage of mainstream LLM-enhanced GNNs. See Appendix A.2 for more details.

4.3 Privacy Attack

In this stage, we evaluate privacy risks across three fundamental attack surfaces, namely *link*, *label*, and *membership* inference, covering six representative attack methods in total. These three categories represent prevalent and well-studied privacy attack threats, targeting the exposure of graph structure, node semantics, and training participation, respectively. To ensure a rigorous and comprehensive assessment, the selected attacks span a range of threat model assumptions, from label-aware to fully black-box settings. Brief descriptions are provided below, with full details deferred to Appendix A.3.

- **Link Inference.** We adopt posterior-based link inference attacks (He et al., 2021b) that predict edge existence by measuring pairwise distances between node posteriors, using *Manhattan distance* (`MLA`) and *Euclidean distance* (`ELA`), capturing structural leakage from model outputs.
- **Label Inference.** We consider two label inference attacks of different adversarial strength. *Homophily Guessing Attack* (`HGA`) (Meng et al., 2023) exploits graph homophily (Zhu et al., 2020) by inferring the victim's label via majority vote over neighbor labels. *Node Infiltration Attack* (`NIA`) (Meng et al., 2023) injects a crafted node connected to the victim and infers its label from model posteriors without accessing label information, representing a complementary threat model.
- **Membership Inference.** Two attacks are examined. `MIA` (Olatunji et al., 2021) trains a shadow GNN to mimic the target model and a binary MLP on member and non-member posteriors to infer training membership. `NMA` (He et al., 2021c) extracts top-2 posterior probabilities from 0-hop and 2-hop queries, fuses them through linear layers, and feeds them into a binary MLP classifier.

4.4 Risk Assessment

In this stage, we adopt standard evaluation metrics to assess the effectiveness of the three types of attacks introduced in Sec. 4.3. Specifically, for membership inference attacks, we use *attack accuracy* and *AUC* as the primary metrics. For link inference attacks, we adopt *AUC* to measure the ability to distinguish existing and non-existing edges. For label inference attacks, we use *attack accuracy* to evaluate the recovery of node labels. These metrics capture the extent of privacy leakage from different perspectives at the representation level. In addition, we incorporate model utility into the evaluation by using *node classification accuracy* as the utility metric. We also consider distributional measures such as *cumulative distribution functions (CDFs)* of embeddings for more fine-grained analysis. By jointly analyzing privacy risk and utility, we aim to systematically characterize the trade-off between performance improvement and privacy leakage in LLM-enhanced GNNs.

Table 2: Node classification accuracy (%) across different datasets based on the GCN backbone. The highest results are highlighted with **bold**, while the second-best results are marked with underline.

Dataset	Shallow	TAPE	KEA	LLaMA	Linq	SimTeg	E5-Large
Cora	80.8 ± 1.1	82.1 ± 1.1	83.2 ± 0.8	82.2 ± 0.6	**84.8 ± 0.4**	83.3 ± 0.9	84.5 ± 0.9
Tape-Arxiv23	55.4 ± 0.7	**72.3 ± 0.6**	65.6 ± 0.5	70.3 ± 0.7	71.4 ± 0.4	70.7 ± 0.6	71.3 ± 0.5
CiteSeer	72.3 ± 1.2	73.6 ± 1.6	75.2 ± 1.5	75.3 ± 1.6	**76.6 ± 0.6**	74.9 ± 0.8	76.2 ± 2.1
Ogbn-Products	65.6 ± 0.5	**83.1 ± 0.2**	80.9 ± 0.3	81.2 ± 0.3	81.7 ± 0.8	81.4 ± 0.2	82.2 ± 0.1
Instagram	63.7 ± 0.7	65.6 ± 0.8	66.1 ± 0.6	67.3 ± 0.4	**67.7 ± 0.7**	64.5 ± 0.7	65.6 ± 0.9
Reddit	60.0 ± 0.9	61.5 ± 1.0	61.2 ± 1.0	**62.8 ± 0.3**	62.5 ± 1.1	62.3 ± 0.4	61.1 ± 0.5

4.5 Defense Analysis

In this stage, we evaluate representative differential privacy (DP) (Dwork, 2008) defenses for LLM-enhanced GNNs. DP provides formal privacy guarantees through random noise injection and has been widely adopted in graph learning. Specifically, we consider two complementary privacy settings. *First*, we evaluate several classical local DP (LDP) mechanisms, including 1B (Ding et al., 2017), LP (Phan et al., 2017), AG (Balle & Wang, 2018), SW (Li et al., 2020), MB (Sajadmanesh & Gatica-Perez, 2021), PM (Wang et al., 2019a), and RR (Warner, 1965), which protect sensitive node representations, links, or labels before model training. *Second*, we consider GAP (Sajadmanesh et al., 2023), a graph-specific DP method that protects privacy during GNN training. Details are provided in Appendix A.4. Together, these mechanisms allow us to assess privacy defenses across different protection stages and analyze their privacy-utility trade-offs.

5 Evaluation

In this section, we conduct a comprehensive experimental study to evaluate the privacy risks in LLM-enhanced GNNs. We begin by introducing the experimental setup (▷ Sec. 5.1), followed by detailed results and analyses (▷ Sec. 5.2–Sec. 5.6), structured around the following key questions:

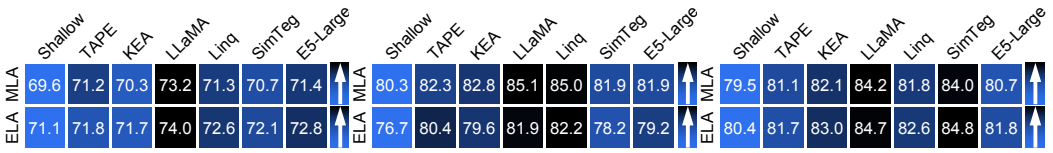
- ◇ **RQ1:** How much do LLM-enhanced GNNs improve utility? (▷ Sec. 5.2)
- ◇ **RQ2:** How vulnerable are LLM-enhanced GNNs to link inference attacks? (▷ Sec. 5.3)
- ◇ **RQ3:** How vulnerable are LLM-enhanced GNNs to label inference attacks? (▷ Sec. 5.4)
- ◇ **RQ4:** How vulnerable are LLM-enhanced GNNs to membership inference attacks? (▷ Sec. 5.5)
- ◇ **RQ5:** How effective are defense mechanisms in mitigating these privacy risks? (▷ Sec. 5.6)

5.1 Experimental Settings

Implementation Details. All experiments are conducted on a machine running Ubuntu 20.04 LTS, equipped with dual Intel® Xeon® Gold 6348 CPUs, 100GB RAM, and an NVIDIA® A800 80GB GPU. GNN models are implemented using PyTorch-Geometric (PyG)[2], while LLMs and text encoders are accessed via official APIs or Hugging Face Transformers (Wolf et al., 2020). Following standard practice (Fu et al., 2025; Ma et al., 2026), each dataset is split into training, validation, and test sets with a ratio of 10%/10%/80%. Hyperparameters for GNN models are selected via grid search to ensure competitive node classification performance. Unless otherwise specified, GCN, SAGE, GAT, and GIN use two message-passing layers with a hidden dimension of 256, a dropout rate of 0.5, and a learning rate of 0.001. For GAT, the first layer employs eight attention heads and the second layer uses a single head. APPNP, SGC, and SSGC follow their propagation-specific configurations detailed in Appendix B.1. To reduce randomness, each experiment is repeated five times with different random seeds, and the reported results correspond to the averaged performance.

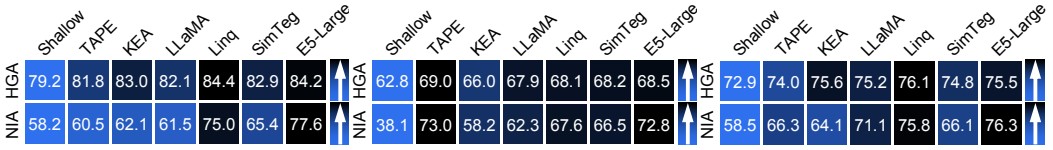
Baseline. To assess whether LLM/LM-augmented node embeddings introduce additional privacy risks, we adopt shallow embeddings constructed using Bag-of-Words (BoW) (Harris, 1954) and Word2Vec (Church, 2017) as the primary baseline. We additionally include a vanilla pretrained LM, *Pure DeBERTa* (He et al., 2021a), which directly encodes the original node text without LLM-generated augmentation, to more clearly disentangle general semantic richness from LLM-specific enhancement. Details of the *Shallow* baseline and the *Pure DeBERTa* comparison are provided in Appendices B.2 and B.9, respectively.

[2] https://www.pyg.org



(a) Cora (b) Tape-Arxiv23 (c) CiteSeer

Figure 3: Link inference results (MLA/ELA, AUC %) on Cora, Tape-Arxiv23, and CiteSeer with GCN. Additional results across more datasets and settings are provided in Appendix B.4.



(a) Cora (b) Tape-Arxiv23 (c) CiteSeer

Figure 4: Label inference results (HGA/NIA, Accuracy %) on Cora, Tape-Arxiv23, and CiteSeer with GCN. Additional results across more datasets and settings are provided in Appendix B.5.

5.2 Utility Evaluation

Before evaluating privacy risks, we first assess whether LLM-enhanced GNNs indeed yield utility gains over shallow text representations. We conduct node classification experiments across all six main datasets, with results reported in Table 2 and full details provided in Appendix B.3. The results show that LLM-enhanced GNNs consistently outperform the *Shallow* baseline in node classification accuracy (%), confirming that semantic enrichment from LLMs/LMs effectively improves node representation quality. These utility gains motivate a deeper investigation into their privacy implications, as stronger representations may simultaneously expose more sensitive information.

5.3 Vulnerability to Link Inference

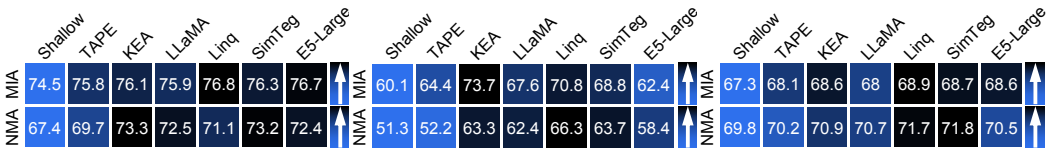
We conduct experiments using two link inference attacks (MLA and ELA), with results reported in Fig. 3 and additional results and analyses provided in Appendix B.4. The results reveal two key observations: ① Compared to the *Shallow* baseline, LLM-enhanced GNNs generally exhibit higher attack AUC (%), indicating that semantic enrichment increases the vulnerability of node representations to link inference attacks. ② Across different LLM-enhanced GNN variants, the attack AUC values remain relatively close to one another, suggesting that the increased privacy vulnerability is a general effect of semantic enrichment rather than being driven by any particular model.

In-depth Analysis. We analyze the distribution of embedding distances between connected and unconnected node pairs on the Tape-Arxiv23 dataset using GCN. As shown in Fig. 6(a), compared to *Shallow*, *Linq* exhibits a pronounced separation between connected and unconnected node pairs: distances between connected pairs decrease while those between unconnected pairs increase, resulting in a larger margin in the embedding space. This increased geometric separability provides a plausible explanation for the higher link-inference attack performance.

5.4 Vulnerability to Label Inference

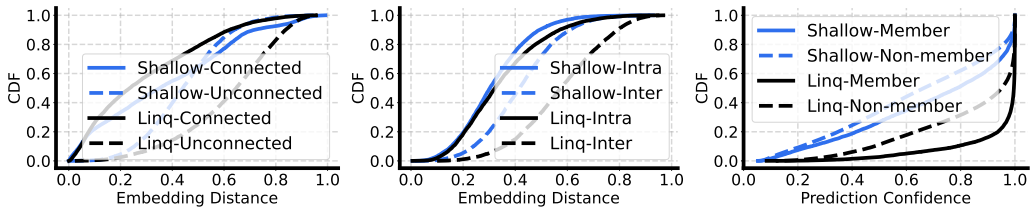
We evaluate vulnerability to label privacy inference using two representative attacks (HGA and NIA), with results shown in Fig. 4 and additional details provided in Appendix B.5. The results reveal two key observations: ① LLM-enhanced GNNs consistently exhibit stronger label leakage (measured by attack accuracy (%)) compared to the *Shallow* baseline, suggesting that incorporating semantic knowledge enables label-related signals to be encoded more explicitly in node representations. ② Performance differences across LLM-enhanced GNN variants remain relatively moderate, indicating that the elevated vulnerability is not driven by any specific model architecture, but rather reflects a common consequence of leveraging semantically enriched features.

In-depth Analysis. We analyze the distribution of embedding distances between *intra-class* (same-label) and *inter-class* (different-label) node pairs on the Tape-Arxiv23 dataset using GCN model. The results, presented as cumulative distribution functions (CDFs) in Fig. 6(b), show that compared to *Shallow*, *Linq* exhibits a more pronounced distributional gap between intra-class and inter-class distances, indicating a clearer class-wise separation in the embedding space. This suggests that LLM-enhanced GNNs learn a more structured, label-aware representation, whose enhanced discriminability, while improving classification utility, simultaneously makes label inference attacks more effective.



(a) Cora (b) Tape-Arxiv23 (c) CiteSeer

Figure 5: Membership inference results (MIA/NMA, AUC/Accuracy %) on Cora, Tape-Arxiv23, and CiteSeer with GCN. Additional results across more datasets and settings are provided in Appendix B.6.



(a) Link Inference (b) Label Inference (c) Membership Inference

Figure 6: In-depth analysis of privacy vulnerability on Tape-Arxiv23 with GCN, comparing *Linq* against the *Shallow* baseline. (a) Embedding distance distributions of connected *vs.* unconnected pairs (link inference). (b) CDF of embedding distances of intra-class *vs.* inter-class pairs (label inference). (c) Prediction confidence distributions of member *vs.* non-member nodes (membership inference).

5.5 Vulnerability to Membership Inference

We evaluate membership privacy vulnerability using two representative attacks, namely MIA (measured by AUC (%)) and NMA (measured by attack accuracy (%)). Results are reported in Fig. 5, with additional analyses provided in Appendix B.6. The results show: ① LLM-enhanced GNNs consistently exhibit higher membership leakage than the *Shallow* baseline across datasets, suggesting that semantic enrichment amplifies the model's tendency to memorize training-specific patterns and leaves stronger membership signals in the learned representations. ② Among different LLM-enhanced GNN variants, the degree of membership leakage above the *Shallow* baseline varies across methods, reflecting that different feature enhancers encode training membership signals to different extents.

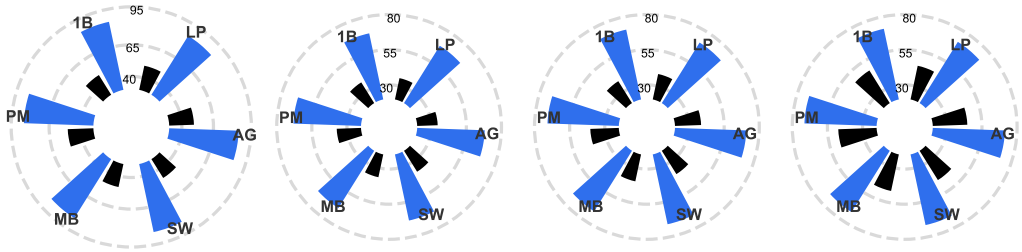
In-depth Analysis. Membership inference attacks fundamentally exploit the behavioral discrepancy between member and non-member samples. To investigate this, we compare the prediction confidence distributions of member and non-member nodes on Tape-Arxiv23 using GCN, as shown in Fig. 6(c). Compared to *Shallow*, *Linq* exhibits a more pronounced distributional gap between member and non-member confidence scores, indicating that LLM-enhanced representations encode stronger training-specific signals. This makes the behavioral discrepancy between member and non-member nodes more distinguishable, thereby amplifying the risk of membership inference attacks.

5.6 Effectiveness of Defenses

In this section, we evaluate representative DP defenses under both input-level perturbation and private GNN training settings. The main-text experiments focus on embedding-, link-, and label-level LDP on Tape-Arxiv23 using GCN with Linq as the feature enhancer. We further evaluate GAP-based private GNN training and cross-backbone defense generalization in Apps B.12 and B.13, respectively.

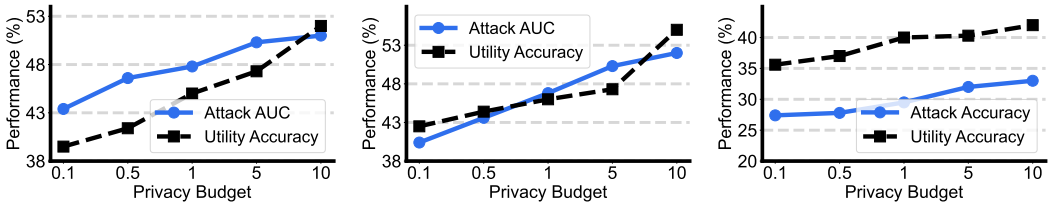
Embedding DP. We consider six local DP mechanisms for node embedding perturbation (1B, LP, AG, SW, MB, PM), and evaluate their effectiveness against MLA, NIA, and MIA for link, label, and membership inference, respectively, on Tape-Arxiv23 using GCN with Linq as the feature enhancer. The results are shown in Fig. 7. Under a moderate privacy budget ($\epsilon_e = 5.0$), Fig. 7(a)-(c) show that embedding perturbation can mitigate all three types of attacks to varying degrees. However, Fig. 7(d) reveals a significant drop in model utility, indicating a clear privacy-utility trade-off. We further examine the effect of varying the privacy budget $\epsilon_e \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$ (under the MB) on MLA (Fig. 8(a)). As ϵ_e decreases, the performance of the fixed MLA attack and model utility both decline, indicating a privacy-utility trade-off under the evaluated attack protocol.

Link DP. We further study link-level local DP using the RR. As shown in Fig. 8(b), applying RR to the graph structure reduces the effectiveness of MLA. Nevertheless, it simultaneously incurs noticeable utility degradation, indicating a poor privacy-utility trade-off for structure-level protection.



(a) Link Attack (b) Label Attack (c) Membership Attack (d) Utility Accuracy

Figure 7: Effectiveness of six local embedding DP mechanisms under a fixed privacy budget $\epsilon_e = 5.0$ on Tape-Arxiv23 with GCN and Linq; blue bars denote results before perturbation. (a)-(c) show attack performance for MLA, NIA, and MIA, respectively. (d) reports the corresponding model utility.



(a) Embedding-level DP (b) Link-level DP (c) Label-level DP

Figure 8: Attack and utility performance under varying privacy budgets on Tape-Arxiv23 with GCN and Linq. (a) Embedding-level DP (ϵ_e) *vs.* MLA (AUC %) and utility (Accuracy %). (b) Link-level DP (RR) *vs.* MLA (AUC %) and utility. (c) Label-level DP (k-RR) *vs.* HGA (Accuracy %) and utility.

Label DP. We examine label-level DP using the k-RR mechanism against HGA-based label inference. As shown in Fig. 8(c), while label perturbation partially reduces attack performance, it inevitably weakens the learning of true label distributions, leading to degraded classification accuracy. This highlights the inherent tension between protecting label privacy and maintaining model utility.

6 Related Work

In this section, we provide a brief overview of *LLM-enhanced GNNs* and *privacy attacks on GNNs*. A more comprehensive discussion and additional references are provided in Appendix C.

LLM-Enhanced GNNs. Recent studies have explored integrating LLMs with GNNs to encode rich textual attributes, giving rise to *LLM-enhanced GNNs* (He et al., 2024; Chen et al., 2024; Zhu et al., 2024; Chien et al., 2022; Zhang et al., 2025b), which significantly improve node representations and model utility. Existing methods fall into two categories: *explanation-based methods*, which use LLMs to generate semantic descriptions that are subsequently encoded by a language model (He et al., 2024); and *embedding-based methods*, which directly encode textual attributes via LLMs or fine-tuned language models (Duan et al., 2023). While these approaches have demonstrated significant utility gains, the privacy implications of incorporating semantically enriched representations into graph learning remain largely unexplored, motivating our systematic investigation.

Privacy Attacks on GNNs. Privacy risks in graph learning are commonly studied under three threat scenarios. Membership inference attacks (Olatunji et al., 2021; Guan et al., 2025; He et al., 2021c) determine whether a node was used for training. Link inference attacks (Wu et al., 2022a; He et al., 2021b) recover sensitive graph structure from node representations. Label inference attacks (Meng et al., 2023) infer private node labels. Prior work mainly studies these threats in conventional GNNs, while their behavior under semantically enriched LLM-based representations remains underexplored.

7 Conclusion

In this work, we systematically audit privacy risks in LLM-enhanced GNNs across link, label, and membership inference. Extensive experiments across ten TAG datasets and diverse model configurations show that LLM-enhanced GNNs generally exhibit greater empirical privacy vulnerability than shallow baselines, accompanied by more distinguishable privacy-sensitive signals. We further show that DP can mitigate these risks, but often at the cost of model utility. Overall, our findings provide practical insights for developing more secure and trustworthy graph learning systems.

AI USE STATEMENT

This work studies privacy risks in LLM-enhanced graph neural networks, where LLMs serve as semantic feature enhancers within the evaluated victim models. Separately, LLM-based tools were used only for limited language polishing and writing refinement during manuscript preparation. They were not involved in research problem formulation, methodological or experimental design, theoretical development, result interpretation, or scientific conclusions. All scientific contributions were developed and validated by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

This work investigates privacy risks in LLM-enhanced graph neural networks and evaluates representative privacy inference attacks and defense mechanisms on publicly available benchmark datasets. The study aims to better understand privacy vulnerabilities and mitigation strategies, ultimately promoting safer, more privacy-preserving, and trustworthy graph learning systems.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main paper describes the privacy auditing framework, threat model, experimental setup, evaluation metrics, and defense settings. App. A provides detailed descriptions of the datasets, feature enhancement methods, GNN backbones, privacy attacks, and defense mechanisms, including the core prompts and attack procedures. App. B reports complete and additional experimental results, including cross-domain evaluation, recent LLM backbones, controlled analyses, cross-backbone studies, and private GNN training. We also provide the processed datasets and source code through our anonymous repository: Anonymous link.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in Neural Information Processing Systems (NeurIPS)*, 26:2814–2822, 2013.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, pp. 394–403, 2018.

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

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems (NeurIPS)*, 37:96640–96670, 2024.

Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S. Dhillon. Node feature extraction by self-supervised multi-scale neighborhood prediction. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy-yong Sohn. Linq-embed-mistral technical report. *arXiv preprint arXiv:2412.03223*, 2024.

Kenneth Ward Church. Word2vec. *Natural Language Engineering (NLEJ)*, 23(1):155–162, 2017.

Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pp. 1655–1658, 2018.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the world wide web. *AAAI/IAAI*, 3 (3.6):2, 1998.

Wei-Yen Day, Ninghui Li, and Min Lyu. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)*, pp. 123–138, 2016.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:3571–3580, 2017.

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

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 429–438, 2013.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research (JMLR)*, 24(43):1–48, 2023.

Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation (TAMC)*, pp. 1–19, 2008.

Jie Fu, Yuan Hong, Zhili Chen, and Wendy Hui Wang. Safeguarding graph neural networks against topology inference attacks. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2144–2158, 2025.

David F Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.

Google. Vertex ai - google cloud. https://cloud.google.com/vertex-ai, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Faqian Guan, Tianqing Zhu, Hanjin Tong, and Wanlei Zhou. Attention-based membership inference attacks on graph neural network through topological features. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 22(6):6469–6486, 2025.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:1024–1034, 2017.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Longzhu He, Chaozhuo Li, Peng Tang, and Sen Su. Going deeper into locally differentially private graph neural networks. In *Forty-second International Conference on Machine Learning (ICML)*, pp. 22548–22565, 2025.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations (ICLR)*, 2021a.

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

Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *30th USENIX Security Symposium (USENIX Security)*, pp. 2669–2686, 2021b.

Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429*, 2021c.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:22118–22133, 2020.

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

IBM. Infosphere virtual data pipeline — ibm. https://www.ibm.com/products/ibm-infosphere-virtual-data-pipeline, 2018.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning (ICML)*, pp. 2436–2444, 2016a.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *Journal of Machine Learning Research (JMLR)*, 17(17):1–51, 2016b.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing (SICOMP)*, 40(3):793–826, 2011.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*, pp. 1–15, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations (ICLR)*, pp. 1–15, 2019.

Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. Natural language understanding and inference with mllm in visual question answering: A survey. *ACM Computing Surveys (CSUR)*, 57(8):1–36, 2025.

Andrei Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

Juanhui Li, Harry Shomer, Haitao Mao, Shenglai Zeng, Yao Ma, Neil Shah, Jiliang Tang, and Dawei Yin. Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:3853–3866, 2023.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 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)*, pp. 8123–8131, 2024.

Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Škoric. Estimating numerical distributions under local differential privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 621–635, 2020.

Wanyu Lin, Baochun Li, and Cong Wang. Towards private learning on decentralized graphs with local differential privacy. *IEEE Transactions on Information Forensics and Security (TIFS)*, 17: 2936–2946, 2022.

Shuwen Liu, Bernardo Grau, Ian Horrocks, and Egor Kostylev. Indigo: Gnn-based inductive knowledge graph completion using pair-wise encoding. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:2034–2045, 2021.

Renze Lou, Kai Zhang, and Wenpeng Yin. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095, 2024.

Yuhang Ma, Jie Wang, and Zheng Yan. Are llm-enhanced graph neural networks robust against poisoning attacks? *arXiv preprint arXiv:2603.26105*, 2026.

Seiji Maekawa, Koki Noda, Yuya Sasaki, et al. Beyond real-world benchmark datasets: An empirical study of node classification with gnns. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:5562–5574, 2022.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Lingshuo Meng, Yijie Bai, Yanjiao Chen, Yutong Hu, Wenyuan Xu, and Haiqin Weng. Devil in disguise: Breaching graph neural networks privacy through infiltration. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1153–1167, 2023.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems (NeurIPS)*, 26:3111–3119, 2013.

Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pp. 11–20, 2021.

Parlai. Parlai. https://ai.facebook.com/tools/parlai, 2017.

NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 385–394, 2017.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024 (ACL Findings)*, pp. 13025–13048, 2024.

Leonardo F. R. Ribeiro, Pedro H. P. Savarese, and Daniel R. Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 385–394, 2017.

Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2130–2145, 2021.

Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. {GAP}: Differentially private graph neural networks with aggregation perturbation. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 3223–3240, 2023.

Entong Shen and Ting Yu. Mining frequent graph patterns with differential privacy. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 545–553, 2013.

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

Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning (ICML)*, pp. 9583–9592, 2020.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations (ICLR)*, pp. 1–15, 2018.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649, 2019a.

Tianhao Wang, Bolin Ding, Jingren Zhou, Cheng Hong, Zhicong Huang, Ninghui Li, and Somesh Jha. Answering multi-dimensional analytical queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data (SIGMOD)*, pp. 159–176, 2019b.

Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association (JASA)*, 60(309):63–69, 1965.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pp. 38–45, 2020.

Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *2022 IEEE Symposium on Security and Privacy (S&P)*, pp. 2005–2024, 2022a.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning (ICML)*, pp. 6861–6871, 2019.

Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (S&P)*, pp. 304–317, 2020a.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys (CSUR)*, 55(5):1–37, 2022b.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 32(1):4–24, 2020b.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. Mapping the minds of llms: A graph-based analysis of reasoning llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 17762–17774, 2025.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations (ICLR)*, pp. 1–15, 2019.

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

Chengdong Yang, Hongrui Liu, Daixin Wang, Zhiqiang Zhang, Cheng Yang, and Chuan Shi. Flag: Fraud detection with llm-enhanced graph neural network. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2 (SIGKDD)*, pp. 5150–5160, 2025.

Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning (ICML)*, pp. 40–48, 2016.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7370–7377, 2019.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems. *ACM Computing Surveys (CSUR)*, 58(6):1–38, 2025.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025a.

Yi Zhang, Yuying Zhao, Zhaoqing Li, Xueqi Cheng, Yu Wang, Olivera Kotevska, Philip S Yu, and Tyler Derr. A survey on privacy in graph neural networks: Attacks, preservation, and applications. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 36(12):7497–7515, 2024.

Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. In *31st USENIX Security Symposium (USENIX Security)*, pp. 4543–4560, 2022.

Zihao Zhang, Xunkai Li, Rong-Hua Li, Bing Zhou, Zhenjun Li, and Guoren Wang. Toward general and robust llm-enhanced text-attributed graph learning. *arXiv preprint arXiv:2504.02343*, 2025b.

Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *9th International Conference on Learning Representations (ICLR)*, pp. 1–15, 2021.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:7793–7804, 2020.

Xiaochen Zhu, Vincent YF Tan, and Xiaokui Xiao. Blink: Link local differential privacy in graph neural networks via bayesian estimation. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2651–2664, 2023.

Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. Efficient tuning and inference for large language models on textual graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5734–5742, 2024.

APPENDIX

A Implementation Details **17**

A.1 Dataset 17

A.2 Victim Model 17

A.3 Privacy Attack 18

A.4 Defense Analysis 19

B Additional Experimental Results **22**

B.1 GNN Backbone 22

B.2 Baseline 23

B.3 Additional Results on Utility Evaluation 23

B.4 Additional Results on Link Inference 26

B.5 Additional Results on Label Inference 26

B.6 Additional Results on Membership Inference 26

B.7 Evaluation on Additional Graph Domains 26

B.8 Evaluation with More Recent LLM Backbones 27

B.9 Comparison with a Vanilla Pretrained LM 28

B.10 Analysis of Potential Confounding Factors 28

B.11 Cross-Backbone Mechanistic Analysis 29

B.12 Private GNN Training with GAP 30

B.13 Defense Generalization across GNN Backbones 30

C Related Work **31**

C.1 Graph Neural Networks 31

C.2 LLM-Enhanced GNNs 31

C.3 Privacy Attacks on GNNs 32

D Limitation & Broader Impact **32**

D.1 Limitation 32

D.2 Broader Impact 32

A Implementation Details

In this section, we present the details of our privacy auditing framework, including dataset (▷ Sec. A.1), victim model (▷ Sec. A.2), privacy attack (▷ Sec. A.3), and defense analysis (▷ Sec. A.4)

A.1 Dataset

In this work, we consider 10 text-attributed graph datasets spanning diverse real-world domains. Our main evaluation uses six representative datasets from citation networks, social networks, and e-commerce platforms, whose statistics are summarized in Table 1 and described below. We further extend the evaluation to four datasets from Wikipedia, transportation, and university website domains; their descriptions and experimental results are provided in Appendix B.7.

- ◇ *Cora* (McCallum et al., 2000). A widely used citation network where nodes represent scientific publications and edges denote citation relationships. Each node is associated with a bag-of-words feature vector derived from paper content, and the task is to classify papers into 7 research topics.
- ◇ *CiteSeer* (Yang et al., 2016). A citation graph consisting of scientific articles linked via citation edges, with node features extracted from paper abstracts and titles across 6 publication categories. Compared to Cora, CiteSeer exhibits sparser connectivity and a higher proportion of isolated nodes.
- ◇ *Tape-Arxiv23* (He et al., 2024). A citation graph constructed from arXiv computer science papers, with node features derived from LLM-generated textual descriptions of titles and abstracts. Its rich representations make it well-suited for evaluating LLM-enhanced graph learning.
- ◇ *Instagram* (Huang et al., 2024). A social network where nodes represent users and edges capture interaction relationships, with node features constructed from user-generated textual content. Labels correspond to user categories, and the sensitive nature of social content makes this dataset particularly relevant for privacy risk evaluation.
- ◇ *Reddit* (Huang et al., 2024). A social graph where nodes represent posts and edges connect posts sharing commenters. Node features are derived from each post's textual content.
- ◇ *Ogbn-Products* (He et al., 2024). An e-commerce co-purchase network from the Open Graph Benchmark (OGB) (Hu et al., 2020), where nodes represent Amazon products and edges indicate co-purchase relationships across 47 product categories. Node features are derived from product titles and descriptions, providing rich semantic information for downstream graph learning.

Table 3: Parameters and vendors of the LLMs/LMs underlying the six feature enhancement methods in our main benchmark.

Model Type	LLMs/LMs	#Parameters	Vendor
API-based	GPT-3.5-Turbo	–	OpenAI
Open-source	DeBERTa-base	140M	Microsoft
	E5-Large	300M	Microsoft
	LLaMA-2-7b-hf	7B	Meta
	Linq-Embed-Mistral	7B	LinqAlpha

A.2 Victim Model

In this work, we consider a diverse set of victim model configurations to comprehensively evaluate inference privacy risks across different LLM-enhanced GNN paradigms. Specifically, we combine six LLM/LM-based feature enhancement methods with seven GNN architectures, yielding 42 LLM-enhanced GNN configurations.

A.2.1 LLMs/LMs as Feature Enhancers

We consider six LLM-enhanced node feature enhancement methods, namely *TAPE* (He et al., 2024), *KEA* (Chen et al., 2024), *LLaMA* (Touvron et al., 2023), *Linq* (Choi et al., 2024), *SimTeg* (Duan et al., 2023), and *E5-Large* (Wang et al., 2022), which collectively leverage five representative LLMs/LMs

as their underlying text encoders. Specifically, these underlying models span both API-based and open-source categories, as summarized in Table 3. The API-based model includes *GPT-3.5-Turbo* from OpenAI, which is accessed via a remote inference interface without exposing model weights. The open-source models include *DeBERTa-base* (He et al., 2021a) (140M parameters) from Microsoft, *E5-Large* (Wang et al., 2022) (300M parameters) from Microsoft, *LLaMA-2-7b-hf* (Touvron et al., 2023) (7B parameters) from Meta, and *Linq-Embed-Mistral* (Choi et al., 2024) (7B parameters) from LinqAlpha. These models vary substantially in scale, architecture, and training objectives, enabling us to examine how different levels of semantic richness and model capacity affect downstream privacy leakage. To examine whether our findings remain valid with more recent language models, we additionally evaluate *Llama-3.1-8B* (Grattafiori et al., 2024), *Qwen3-Embedding-8B* (Zhang et al., 2025a), and *DeepSeek-V4-Pro* (DeepSeek-AI, 2026); their configurations and experimental results are provided in Appendix B.8. The core prompt templates used by the two explanation-based methods, *TAPE* and *KEA*, are provided below:

Prompt for TAPE

Input
Title: `[TITLE]`
Text: `[TEXT]`
Candidate Categories: `[LABEL SET]`

Instruction
Based on the title and text above, predict the most relevant categories from the candidate set. If multiple categories are applicable, rank them from most to least likely. For each predicted category, briefly explain which information in the text supports the prediction.

Output Format
Predictions: `[RANKED LABELS]`
Explanation: `[EXPLANATION]`

Prompt for KEA

Input
Text: `[TEXT]`

Instruction
Identify the important domain-specific entities, concepts, or technical terms appearing in the text. For each identified item, provide a concise factual description that explains its meaning in the context of the input.

Output Format

```
{
  "term 1":  "description 1",
  "term 2":  "description 2",
  ...
}
```

Output only the dictionary.

A.2.2 GNN Architectures as Downstream Backbones

We consider seven GNN architectures (*GCN* (Kipf & Welling, 2017), *SAGE* (Hamilton et al., 2017), *GAT* (Velickovic et al., 2018), *GIN* (Xu et al., 2019), *APPNP* (Klicpera et al., 2019), *SGC* (Wu et al., 2019), and *SSGC* (Zhu & Koniusz, 2021)) as downstream backbones, whose detailed descriptions and implementation configurations are provided in Appendix B.1.

A.3 Privacy Attack

In this paper, we evaluate privacy risks across three fundamental attack surfaces, namely *link*, *label*, and *membership* inference, covering six attacks in total. For reproducibility, we provide the detailed procedures of the six attacks in Algorithms 1, 2, and 3. The same attack implementations, adversarial knowledge, query access, and evaluation protocols are applied to both Shallow and LLM-enhanced GNNs. More details are provided below:

- **Link Inference.** We adopt posterior-based link inference attacks (He et al., 2021b) that predict edge existence by measuring pairwise distances between node posteriors output by the target GNN.

Specifically, we consider two distance metrics: *Manhattan distance* (MLA) and *Euclidean distance* (ELA). A smaller distance between two nodes' posteriors indicates a higher likelihood of an edge existing between them, enabling the attacker to infer graph structure directly from model outputs without access to model parameters or training data.

- ◇ **Label Inference.** We consider two label inference attacks of different adversarial strength. *Homophily Guessing Attack* (HGA) (Meng et al., 2023) exploits the homophily property of graphs (Zhu et al., 2020), which states that connected nodes tend to share the same label. It infers the victim node's label via majority voting over the labels of its neighbors, requiring access to partial label information as prior knowledge. *Node Infiltration Attack* (NIA) (Meng et al., 2023) takes a stronger approach by injecting a carefully crafted node connected to the victim into the graph. The attacker then queries the target model and infers the victim's label purely from the resulting posterior probabilities, requiring no access to any ground-truth label information. Compared with HGA, NIA requires no access to ground-truth label information but assumes an active node-injection capability, representing a complementary and more aggressive threat model.
- ◇ **Membership Inference.** Two representative membership inference attacks are examined. MIA (Olatunji et al., 2021) trains a shadow GNN to mimic the target model and queries member and non-member shadow nodes to construct posterior-based attack samples with binary membership labels. A binary MLP is then trained to infer whether a target node belongs to the training set. NMA (He et al., 2021c) trains a shadow GNN to approximate the behavior of the target model, extracting the top-2 posterior probabilities under 0-hop (node-only) and 2-hop (neighborhood subgraph) queries as two feature sets, which are fused via independent linear layers and fed into a binary MLP to infer the membership status of any target node.

Algorithm 1 Posterior-Based Link Inference (MLA/ELA)

Require: Node posteriors $\{\mathcal{R}_v\}$, candidate pairs $\mathcal{P}$
Ensure: Predicted links
$\mathbf{D}_{\text{MLA}} \leftarrow \textsc{ManhattanDistance}(\mathcal{P}, \{\mathcal{R}_v\})$ {*Compute ℓ_1 posterior distances*}
$\mathbf{D}_{\text{ELA}} \leftarrow \textsc{EuclideanDistance}(\mathcal{P}, \{\mathcal{R}_v\})$ {*Compute ℓ_2 posterior distances*}
for $\mathbf{D} \in \{\mathbf{D}_{\text{MLA}}, \mathbf{D}_{\text{ELA}}\}$ **do**
  $\mathbf{C} \leftarrow \textsc{KMeans}(\mathbf{D}, K = 2)$ {*Separate linked and unlinked pairs*}
  Mark the cluster with the smaller centroid as linked
end for
return Predicted link labels

Algorithm 2 Label Inference Attacks (HGA/NIA)

Require: Target GNN f_θ, graph $\mathcal{G}$, target node v
Ensure: Predicted labels
$\hat{y}_v^{\text{HGA}} \leftarrow \textsc{MajorityVote}(\textsc{NeighborLabels}(v))$ {*Infer from neighboring labels*}
Inject an attacker-controlled node connected to v
$\mathcal{R}_v \leftarrow f_\theta(v)$ {*Query the target posterior*}
if $\mathcal{R}_v$ is sufficiently confident **then**
  $\hat{y}_v^{\text{NIA}} \leftarrow \arg\max_c \mathcal{R}_{v,c}$
else
  $\mathcal{C}_v \leftarrow \textsc{CandidateClasses}(\mathcal{R}_v)$ {*Select plausible labels*}
  $\mathbf{s} \leftarrow \textsc{PerturbAndScore}(f_\theta, v, \mathcal{C}_v)$ {*Estimate label sensitivity*}
  $\hat{y}_v^{\text{NIA}} \leftarrow \arg\max_{c \in \mathcal{C}_v} s_c$
end if
return $\hat{y}_v^{\text{HGA}}, \hat{y}_v^{\text{NIA}}$

A.4 Defense Analysis

In this section, we provide further details on the defense mechanisms evaluated in this work, including an introduction to DP and the specific implementations of embedding-, link- and label-level DP.

Algorithm 3 Membership Inference Attacks (MIA/NMA)

Require: Target GNN f_θ, shadow GNN f_ϕ, shadow nodes $\mathcal{V}_s$, target nodes $\mathcal{V}_q$
Ensure: Membership predictions
$\mathbf{X}_{\text{MIA}} \leftarrow \text{PosteriorFeatures}(f_\phi, \mathcal{V}_s)$ {*Extract posterior-based features*}
$g_{\text{MIA}} \leftarrow \text{TrainAttackClassifier}(\mathbf{X}_{\text{MIA}})$ {*Train binary membership classifier*}
$\mathbf{X}_{\text{NMA}} \leftarrow \text{NeighborhoodFeatures}(f_\phi, \mathcal{V}_s)$ {*Fuse 0-hop and 2-hop posteriors*}
$g_{\text{NMA}} \leftarrow \text{TrainAttackClassifier}(\mathbf{X}_{\text{NMA}})$ {*Train neighborhood-aware classifier*}
for $v \in \mathcal{V}_q$ **do**
  $\hat{m}_v^{\text{MIA}} \leftarrow g_{\text{MIA}}(\text{PosteriorFeatures}(f_\theta, v))$
  $\hat{m}_v^{\text{NMA}} \leftarrow g_{\text{NMA}}(\text{NeighborhoodFeatures}(f_\theta, v))$
end for
return Membership predictions

A.4.1 Differential Privacy

Differential Privacy (DP) (Dwork, 2008) is a rigorous mathematical framework for privacy protection, and has been widely adopted across a broad range of domains, including machine learning (Wu et al., 2020a; Abadi et al., 2016; Triastcyn & Faltings, 2020), data publishing (Duchi et al., 2013; Kairouz et al., 2016b;a; Cormode et al., 2018; Wang et al., 2019a;b), and graph analytics (Sajadmanesh & Gatica-Perez, 2021; Day et al., 2016; Shen & Yu, 2013). The core idea of DP is to bound the statistical influence of any single individual's data on the output of a randomized mechanism, thereby providing formal and quantifiable privacy guarantees. The definition is as follows:

Definition 1 (**(ϵ, δ)-DP**) *A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets $\mathcal{D}, \mathcal{D}'$ differing in one record, and any measurable output set $\mathcal{S}$:*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^\epsilon \cdot \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta, \tag{2}$$

where $\epsilon > 0$ is the privacy budget and $\delta \geq 0$ is a relaxation parameter. When $\delta = 0$, the mechanism satisfies pure ϵ-DP.

While standard DP assumes the existence of a trusted data curator, this assumption is often unrealistic in decentralized settings. *Local DP* (LDP) (Kasiviswanathan et al., 2011; Cormode et al., 2018) addresses this by requiring each user to perturb their own data locally before any transmission, eliminating reliance on a trusted third party.

Definition 2 (**ϵ-LDP**) *A randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Z}$ satisfies ϵ-local differential privacy (ϵ-LDP) if for any two inputs $x, x' \in \mathcal{X}$ and any output $z \in \mathcal{Z}$:*

$$\Pr[\mathcal{M}(x) = z] \leq e^\epsilon \cdot \Pr[\mathcal{M}(x') = z], \quad \epsilon > 0. \tag{3}$$

The privacy budget ϵ governs the privacy-utility trade-off: a smaller ϵ enforces stronger privacy at the cost of greater noise injection and reduced utility. In the context of graph learning, LDP provides a natural and practical privacy model, as node features, graph structure, and labels can each be perturbed locally, making it well-suited for the three defense strategies considered in this work.

A.4.2 Embedding-level DP

To protect node embeddings against the three privacy attacks considered in this work, we consider six LDP mechanisms for embedding perturbation. The first three, namely the *1-bit mechanism* (1B) (Sajadmanesh & Gatica-Perez, 2021), the *Laplace mechanism* (LP) (Phan et al., 2017), and the *Analytic Gaussian mechanism* (AG) (Balle & Wang, 2018), are commonly used for perturbing scalar or vector-valued data under strong local privacy guarantees. Specifically:

- *1-bit Mechanism* (Sajadmanesh & Gatica-Perez, 2021). Perturbs a one-dimensional input $x \in [\alpha, \beta]$ by mapping it probabilistically to either $+1$ or -1, the probability density function is as follows:

$$\Pr[x' = c \mid x] = \begin{cases} \frac{1}{e^\epsilon+1} + \frac{x-\alpha}{\beta-\alpha} \cdot \frac{e^\epsilon-1}{e^\epsilon+1}, & \text{if } c = 1 \\ \frac{e^\epsilon}{e^\epsilon+1} - \frac{x-\alpha}{\beta-\alpha} \cdot \frac{e^\epsilon-1}{e^\epsilon+1}, & \text{if } c = -1 \end{cases}. \tag{4}$$

◇ *Laplace Mechanism* (Phan et al., 2017). Adds Laplace noise to the original input:

$$x' = x + \mathrm{Lap}\left(2/\epsilon\right). \tag{5}$$

◇ *Analytic Gaussian Mechanism (AG)* (Balle & Wang, 2018). The perturbed value is sampled as

$$x' = x + \mathcal{N}(0, \sigma^2), \tag{6}$$

where σ is calibrated according to the analytic Gaussian mechanism to satisfy the target (ϵ, δ)-LDP guarantee after bounding the input domain.

For d-dimensional embeddings, 1B and LP perturb each dimension independently with budget ϵ/d, ensuring overall ϵ-LDP. AG is separately calibrated to satisfy the target (ϵ, δ)-LDP guarantee. The remaining three mechanisms (*piecewise mechanism* (PM) (Wang et al., 2019a), *multi-bit mechanism* (MB) (Sajadmanesh & Gatica-Perez, 2021), and *square wave mechanism* (SW) (Li et al., 2020)) are designed specifically for high-dimensional node features and adopt a *random-then-perturb* paradigm to balance privacy and utility: m dimensions are randomly selected, each perturbed with budget ϵ/m, while the remaining $d - m$ dimensions are zeroed. The details are as follows:

◇ *Piecewise Mechanism* (Wang et al., 2019a). For input $x \in [\alpha, \beta]$, the perturbed value x' is sampled from an expanded domain $[-\mathcal{B}, \mathcal{B}]$ where $\mathcal{B} = \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$:

$$\Pr[x' = c \mid x] = \begin{cases} p, & \text{if } c \in [l(x), r(x)] \\ p/e^{\epsilon}, & \text{if } c \in [-\mathcal{B}, l(x)) \cup (r(x), \mathcal{B}] \end{cases}, \tag{7}$$

where $p = \frac{e^{\epsilon}-e^{\epsilon/2}}{2e^{\epsilon/2}+2}$, $l(x) = \frac{\mathcal{B}+1}{2}x - \frac{\mathcal{B}-1}{2}$, and $r(x) = l(x) + \mathcal{B} - 1$.

◇ *Multi-bit Mechanism* (Sajadmanesh & Gatica-Perez, 2021). Outputs a binary value $x' \in \{-1, 1\}$ for input $x \in [\alpha, \beta]$:

$$\Pr[x' = c \mid x] = \begin{cases} \frac{1}{e^{\epsilon}+1} + \frac{x-\alpha}{\beta-\alpha} \cdot \frac{e^{\epsilon}-1}{e^{\epsilon}+1}, & \text{if } c = 1 \\ \frac{e^{\epsilon}}{e^{\epsilon}+1} - \frac{x-\alpha}{\beta-\alpha} \cdot \frac{e^{\epsilon}-1}{e^{\epsilon}+1}, & \text{if } c = -1 \end{cases}. \tag{8}$$

◇ *Square Wave Mechanism* (Li et al., 2020). Perturbs input $x \in [\alpha, \beta]$ within an expanded domain $[-b-1, b+1]$, where $b = \frac{\epsilon e^{\epsilon} - e^{\epsilon} + 1}{e^{\epsilon}(e^{\epsilon} - \epsilon - 1)}$:

$$\Pr[x' = c \mid x] = \begin{cases} p, & \text{if } c \in [x-b, x+b] \\ p/e^{\epsilon}, & \text{if } c \in [-b-1, x-b) \cup (x+b, b+1] \end{cases}, \tag{9}$$

where $p = \frac{e^{\epsilon}}{2be^{\epsilon}+2}$.

A.4.3 Link-level DP

To protect graph structural privacy, we adopt the *Randomized Response* (RR) mechanism (Warner, 1965), a classical LDP technique applied to binary edge variables (Zhu et al., 2023; Lin et al., 2022). For each potential edge (u, v), the true edge indicator $a_{uv} \in \{0, 1\}$ is flipped with probability $\frac{1}{e^{\epsilon}+1}$:

$$\Pr[a'_{uv} = c \mid a_{uv}] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+1}, & \text{if } c = a_{uv} \\ \frac{1}{e^{\epsilon}+1}, & \text{if } c \neq a_{uv} \end{cases}, \tag{10}$$

where a smaller ϵ results in more frequent flipping and stronger privacy protection. The perturbed adjacency matrix is then used in place of the original graph structure during training. This mechanism satisfies ϵ-LDP and can be applied independently to each edge without coordination across nodes.

A.4.4 Label-level DP

To protect node label privacy, we employ the *k-ary Randomized Response* (k-RR) mechanism (Kairouz et al., 2016a; Sajadmanesh & Gatica-Perez, 2021; Warner, 1965), which extends the binary RR to a discrete domain of k classes. For a true label $y \in \{1, \ldots, k\}$, the perturbed label y' is sampled as:

$$\Pr[y' = c \mid y] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+k-1}, & \text{if } c = y \\ \frac{1}{e^{\epsilon}+k-1}, & \text{if } c \neq y \end{cases}. \tag{11}$$

As $\epsilon \to \infty$, k-RR recovers the true label with certainty, while smaller ϵ values increasingly randomize the output. The perturbed labels are used as supervision signals during training, introducing deliberate noise that reduces the model's ability to encode true label information in node representations, thereby mitigating label inference risks.

A.4.5 PRIVATE GNN TRAINING WITH GAP

Beyond input-level LDP, we further evaluate private GNN training using GAP (Sajadmanesh et al., 2023) under a centralized DP setting. Unlike LDP, which privatizes embeddings, edges, or labels before they reach the server, GAP assumes a trusted curator and injects calibrated noise during neighborhood aggregation to provide node-level privacy. We denote its privacy budget by ϵ_g. Detailed results under different privacy budgets and matched-budget comparisons between Shallow-GCN and Linq-GCN are reported in Appendix B.12.

B ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide supplementary experimental details and analyses complementing Sec. 5, including GNN backbone settings (▷ Sec. B.1), baseline configurations (▷ Sec. B.2), complete results (▷ Sec. B.3–Sec. B.6), and additional analyses on dataset and model generalization, confounding factors, underlying mechanisms, and privacy defenses (▷ Sec. B.7–Sec. B.13).

B.1 GNN BACKBONE

To provide a comprehensive evaluation of the different GNN backbones, we include seven representative models: *GCN* (Kipf & Welling, 2017), *SAGE* (Hamilton et al., 2017), *GAT* (Velickovic et al., 2018), *GIN* (Xu et al., 2019), *APPNP* (Klicpera et al., 2019), *SGC* (Wu et al., 2019), and *SSGC* (Zhu & Koniusz, 2021). GCN, SAGE, GAT, and GIN adopt two message-passing layers with 256 hidden units, while APPNP, SGC, and SSGC follow their propagation-specific configurations. We use ReLU activation functions (Agarap, 2018) and apply dropout (Baldi & Sadowski, 2013) for regularization. For GAT, the first layer employs eight attention heads, while the second layer uses a single head. For propagation-based variants (*i.e.*, APPNP, SGC, and SSGC), we follow their standard configurations with fixed propagation steps and linear classifiers. All implementations are based on PyTorch with the PyTorch Geometric (PyG[3]) library. Experiments are conducted on a server running Ubuntu 20.04 LTS, equipped with dual Intel® Xeon® Gold 6348 CPUs, 100 GB RAM, and an NVIDIA® A800 GPU. We briefly summarize each model as follows:

- ◇ *GCN*[4] (Kipf & Welling, 2017): Graph Convolutional Networks (GCN) serve as a fundamental graph neural network architecture that implements neighborhood aggregation through spectral graph convolution. By leveraging the normalized graph Laplacian, GCN adopts a layer-wise propagation mechanism to aggregate and transform node features from adjacent nodes.
- ◇ *GraphSAGE (SAGE)*[5] (Hamilton et al., 2017): Graph Sample and Aggregate (GraphSAGE) is an inductive graph neural network that learns node embeddings by sampling and aggregating feature information from local neighborhoods. It supports multiple aggregation functions including mean, LSTM, and pooling, yielding high flexibility for large-scale and dynamic graph scenarios.
- ◇ *GAT*[6] (Velickovic et al., 2018): Graph Attention Networks (GAT) improve message passing by introducing an attention mechanism to learn the importance weights of neighboring nodes. This enables adaptive weighting during aggregation and enhances performance on graphs with noisy or unbalanced neighborhoods.
- ◇ *GIN*[7] (Xu et al., 2019): Graph Isomorphism Networks (GIN) are designed to maximize the discriminative power of graph structure representation and are theoretically proven to be as powerful as the Weisfeiler-Lehman (WL) graph isomorphism test (Leman & Weisfeiler, 1968). By employing MLPs and sum aggregation, GIN effectively captures rich structural information in graphs.

[3] https://www.pyg.org
[4] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.models.GCN.html
[5] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.models.GraphSAGE.html
[6] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GATConv.html
[7] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GINConv.html

◇ *APPNP*[8] (Klicpera et al., 2019): Approximate Personalized Propagation of Neural Predictions (APPNP) decouples feature transformation and propagation based on personalized PageRank (Gleich, 2015). It first uses a shallow neural network for feature transformation, then performs multi-step propagation to enhance long-range information delivery while alleviating over-smoothing.

◇ *SGC*[9] (Wu et al., 2019): Simple Graph Convolution (SGC) simplifies GCN by removing non-linear activations and collapsing weight matrices across layers. This results in a linear and computationally efficient model while maintaining competitive performance.

◇ *SSGC* (Zhu & Koniusz, 2021): Simple Spectral Graph Convolution (SSGC) aggregates multi-hop neighborhood information through a modified Markov diffusion kernel, balancing global and local graph information while retaining a shallow and computationally efficient architecture.

B.2 Baseline

To evaluate whether LLM-enhanced node representations introduce additional privacy risks, we construct a set of shallow text embeddings as a reference baseline to disentangle the impact of semantic enrichment from privacy leakage. Specifically, we consider two widely used traditional text representation methods, *Bag-of-Words (BoW)* (Harris, 1954) and *Word2Vec* (Church, 2017), which are collectively referred to as *Shallow* in our experiments. The construction method and statistics for each dataset are summarized in Table 4. We detail each method as follows:

◇ *Bag-of-Words (BoW)* (Harris, 1954). BoW represents each document as a sparse vector over a fixed vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$, where each dimension corresponds to a unique token. For a document d, the BoW representation $\mathbf{x}_{\text{BoW}} \in \mathbb{R}^{|\mathcal{V}|}$ is defined as: $\mathbf{x}_{\text{BoW}}[i] = \text{count}(w_i, d), w_i \in \mathcal{V}$, where $\text{count}(w_i, d)$ denotes the number of occurrences of word w_i in document d. BoW ignores word order and syntactic structure, capturing only lexical frequency information. Despite its simplicity, BoW has been widely adopted as a strong baseline in text classification and retrieval tasks.

◇ *Word2Vec* (Church, 2017). Unlike the above count-based methods, Word2Vec produces dense, low-dimensional word embeddings by training a shallow neural network to predict contextual word co-occurrences. We adopt the Skip-Gram variant, which learns a word embedding $\mathbf{e}_w \in \mathbb{R}^d$ by maximizing the log-likelihood of observing surrounding context words given a center word:

$$\mathcal{L} = \sum_{w \in \mathcal{V}} \sum_{c \in \text{Context}(w)} \log P(c \mid w; \mathbf{e}_w), \tag{12}$$

where $P(c \mid w)$ is typically approximated via negative sampling. For each node, the document-level representation $\mathbf{x}_{\text{W2V}} \in \mathbb{R}^d$ is obtained by averaging the pre-trained embeddings of all tokens in the corresponding text: $\mathbf{x}_{\text{W2V}} = \frac{1}{|d|} \sum_{w_i \in d} \mathbf{e}_{w_i}$. Word2Vec captures local semantic similarities through distributional co-occurrence statistics, yet fundamentally differs from LLM-based representations in that it lacks deep contextual understanding, cross-sentence reasoning, and world knowledge.

Table 4: Traditional embedding methods used for each dataset.

Dataset	Cora	Tape-Arxiv23	CiteSeer	Ogbn-Products	Instagram	Reddit
Method	BoW	Word2Vec	BoW	BoW	Word2Vec	Word2Vec

B.3 Additional Results on Utility Evaluation

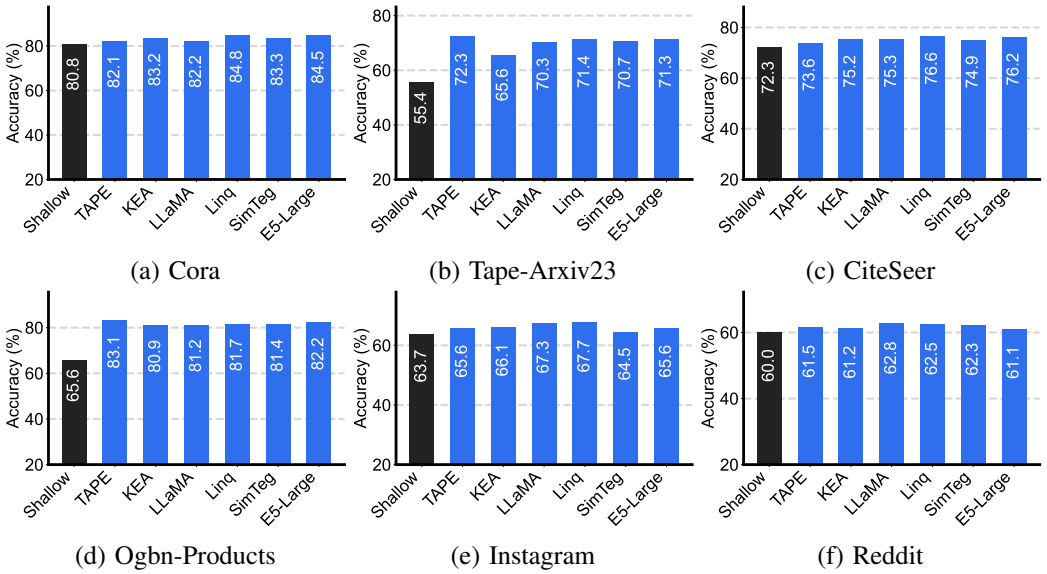
We report the node classification results for SAGE in Table 5, complementing the GCN results presented in the main paper (Table 2). Figures 9 and 10 provide a more intuitive comparison of the performance of the shallow method and the LLM-enhanced method. Consistent with the observations in Sec. 5.2, LLM-enhanced GNNs consistently outperform the *Shallow* baseline across all six datasets, confirming that the utility gains from semantic enrichment generalize across different GNN backbones. These results further validate the representational advantages of LLM-augmented node features and reinforce the motivation for investigating their associated privacy risks.

[8] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.APPNP.html

[9] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.SGConv.html

Table 5: Node classification accuracy (%) across different datasets based on the SAGE backbone. The highest results are highlighted with **bold**, while the second-best results are marked with underline.

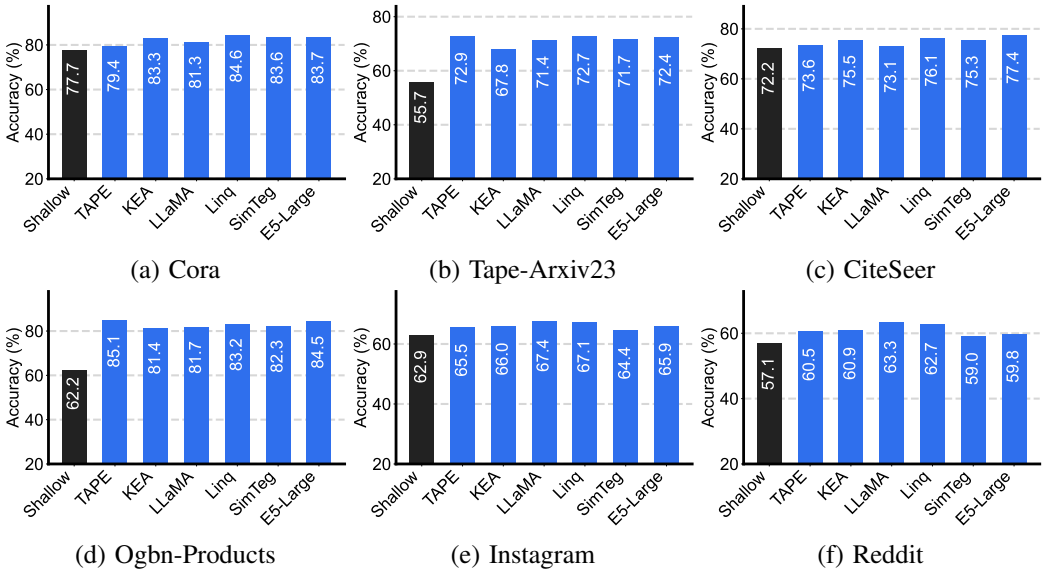
Dataset	Shallow	TAPE	KEA	LLaMA	Linq	SimTeg	E5-Large
Cora	77.7 ± 0.7	79.4 ± 3.5	83.3 ± 1.7	81.3 ± 0.8	**84.6 ± 0.8**	83.6 ± 1.5	83.7 ± 1.5
Tape-Arxiv23	55.7 ± 1.0	**72.9 ± 0.9**	67.8 ± 0.4	71.4 ± 0.3	72.7 ± 0.6	71.7 ± 0.5	72.4 ± 0.2
CiteSeer	72.2 ± 0.7	73.6 ± 2.4	75.5 ± 0.4	73.1 ± 2.5	76.1 ± 0.8	75.3 ± 1.3	**77.4 ± 0.5**
Ogbn-Products	62.2 ± 1.1	**85.1 ± 0.3**	81.4 ± 1.0	81.7 ± 0.5	83.2 ± 0.3	82.3 ± 0.4	84.5 ± 0.6
Instagram	62.9 ± 0.3	65.5 ± 0.4	66.0 ± 0.4	**67.4 ± 0.6**	67.1 ± 0.8	64.4 ± 0.8	65.9 ± 0.7
Reddit	57.1 ± 0.6	60.5 ± 0.5	60.9 ± 0.5	**63.3 ± 0.8**	62.7 ± 0.6	59.0 ± 1.3	59.8 ± 0.6



(a) Cora (b) Tape-Arxiv23 (c) CiteSeer

(d) Ogbn-Products (e) Instagram (f) Reddit

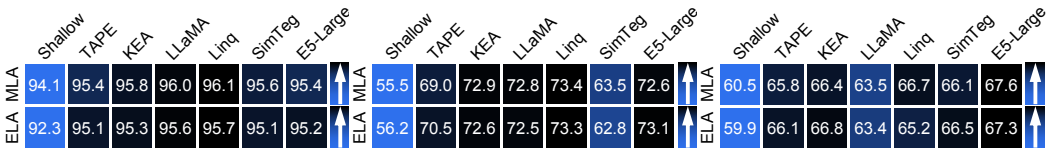
Figure 9: Node classification accuracy (%) across different datasets based on the GCN backbone.



(a) Cora (b) Tape-Arxiv23 (c) CiteSeer

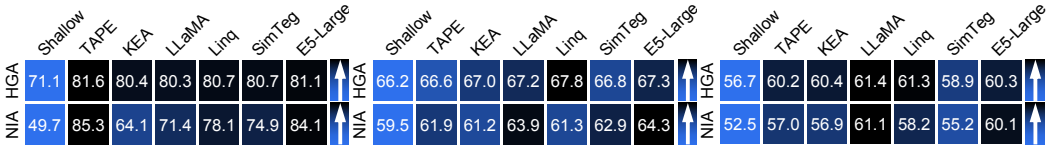
(d) Ogbn-Products (e) Instagram (f) Reddit

Figure 10: Node classification accuracy (%) across different datasets based on the SAGE backbone.



(a) Ogbn-Products (b) Instagram (c) Reddit

Figure 11: Link inference attack (MLA and ELA) results measured by attack AUC (%) across three datasets under GCN, comparing different LLM-based feature enhancers against the *Shallow* baseline.



(a) Ogbn-Products (b) Instagram (c) Reddit

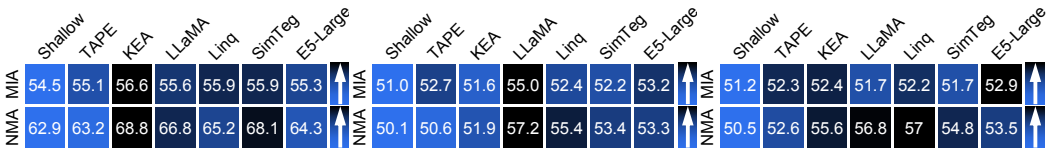
Figure 12: Label inference attack (HGA and NIA) results measured by attack accuracy (%) across three datasets under GCN, comparing different LLM-based feature enhancers against the *Shallow* baseline.

Table 6: MLA attack AUC (%) under different model configurations. The lowest results are highlighted with blue color.

Method	SAGE	GAT	GIN	APPNP	SGC	SSGC
Shallow	72.7 ± 1.3	73.9 ± 1.2	74.6 ± 1.4	72.1 ± 1.1	71.3 ± 1.0	71.8 ± 1.1
TAPE	79.8 ± 2.5	80.9 ± 2.3	81.7 ± 2.4	79.2 ± 2.1	78.5 ± 2.0	79.0 ± 2.1
KEA	82.9 ± 0.7	83.6 ± 0.8	84.5 ± 0.9	82.4 ± 0.7	81.2 ± 0.6	81.8 ± 0.7
LLaMA	82.5 ± 1.0	83.8 ± 1.1	84.1 ± 1.0	82.7 ± 0.9	81.0 ± 0.8	81.6 ± 0.9
Linq	82.7 ± 0.7	83.2 ± 0.8	84.0 ± 0.9	82.9 ± 0.7	81.5 ± 0.6	82.1 ± 0.7
SimTeg	83.3 ± 1.6	84.1 ± 1.5	84.9 ± 1.6	83.0 ± 1.4	82.0 ± 1.3	82.6 ± 1.4
E5-Large	81.3 ± 0.7	82.5 ± 0.8	83.2 ± 0.9	81.7 ± 0.7	80.6 ± 0.6	81.1 ± 0.7

Table 7: NIA attack accuracy (%) under different model configurations. The lowest results are highlighted with blue color.

Method	SAGE	GAT	GIN	APPNP	SGC	SSGC
Shallow	50.5 ± 5.1	51.8 ± 4.8	52.6 ± 5.0	49.9 ± 4.6	48.7 ± 4.2	49.3 ± 4.4
TAPE	58.1 ± 5.4	59.6 ± 5.1	60.8 ± 5.3	57.2 ± 4.8	55.9 ± 4.4	56.6 ± 4.6
KEA	67.0 ± 5.8	68.4 ± 5.6	71.2 ± 5.7	66.5 ± 5.2	64.1 ± 4.7	65.0 ± 4.9
LLaMA	66.7 ± 5.8	69.5 ± 5.4	70.1 ± 5.6	67.8 ± 5.0	63.9 ± 4.8	64.8 ± 4.9
Linq	72.0 ± 4.7	71.2 ± 4.9	73.0 ± 5.1	69.4 ± 4.5	66.8 ± 4.2	67.5 ± 4.3
SimTeg	65.5 ± 4.5	66.2 ± 4.4	67.4 ± 4.6	64.8 ± 4.1	62.9 ± 3.9	63.6 ± 4.0
E5-Large	70.6 ± 5.1	72.4 ± 4.7	72.2 ± 4.9	70.1 ± 4.6	67.3 ± 4.3	68.0 ± 4.4



(a) Ogbn-Products (b) Instagram (c) Reddit

Figure 13: Membership inference attack (MIA and NMA) results measured by AUC (%) or Accuracy (%) under GCN, comparing different LLM-based feature enhancers against the *Shallow* baseline.

Table 8: MIA attack AUC (%) under different GNN model configurations. The lowest results are highlighted with blue color.

Method	SAGE	GAT	GIN	APPNP	SGC	SSGC
Shallow	80.9 ± 1.0	81.6 ± 1.1	82.3 ± 1.2	80.4 ± 1.0	79.8 ± 0.9	80.1 ± 1.0
TAPE	82.1 ± 1.5	83.4 ± 1.4	84.2 ± 1.5	81.9 ± 1.3	81.2 ± 1.2	81.6 ± 1.3
KEA	86.5 ± 1.4	87.2 ± 1.5	88.6 ± 1.6	85.9 ± 1.3	84.7 ± 1.2	85.3 ± 1.3
LLaMA	81.6 ± 0.6	83.8 ± 0.8	84.5 ± 0.9	82.6 ± 0.7	81.0 ± 0.6	81.5 ± 0.7
Linq	83.1 ± 0.6	84.0 ± 0.7	85.2 ± 0.8	83.4 ± 0.6	82.1 ± 0.5	82.7 ± 0.6
SimTeg	84.3 ± 1.3	85.1 ± 1.2	86.0 ± 1.3	83.8 ± 1.1	82.9 ± 1.0	83.3 ± 1.1
E5-Large	82.4 ± 1.4	83.6 ± 1.3	84.8 ± 1.4	82.9 ± 1.2	81.8 ± 1.1	82.2 ± 1.2

B.4 Additional Results on Link Inference

We provide additional link inference results complementing Sec. 5.3. First, Fig. 11 reports the MLA and ELA results under GCN on *Ogbn-Products*, *Instagram*, and *Reddit*, which are moved from the main text for space considerations. Second, we extend the MLA evaluation beyond GCN to six additional GNN architectures, namely *SAGE*, *GAT*, *GIN*, *APPNP*, *SGC*, and *SSGC*, with results summarized in Table 6. Across both additional datasets and GNN architectures, LLM-enhanced GNNs consistently exhibit higher attack AUC (%) than the *Shallow* baseline, further supporting that the elevated link-inference vulnerability is not specific to a particular dataset or backbone.

B.5 Additional Results on Label Inference

We provide additional label inference results complementing Sec. 5.4. First, Fig. 12 reports the HGA and NIA results under GCN on *Ogbn-Products*, *Instagram*, and *Reddit*, which are moved from the main text for space considerations. Second, we extend the NIA evaluation beyond GCN to six additional GNN architectures, namely *SAGE*, *GAT*, *GIN*, *APPNP*, *SGC*, and *SSGC*, with results summarized in Table 7. Across both additional datasets and GNN architectures, LLM-enhanced GNNs consistently exhibit higher label leakage than the *Shallow* baseline, indicating that the observed vulnerability is not specific to a particular dataset or backbone.

B.6 Additional Results on Membership Inference

We provide additional membership inference results complementing Sec. 5.5. First, Fig. 13 reports the MIA and NMA results under GCN on *Ogbn-Products*, *Instagram*, and *Reddit*, which are moved from the main text for space considerations. Second, we extend the MIA evaluation beyond GCN to six additional GNN architectures, namely *SAGE*, *GAT*, *GIN*, *APPNP*, *SGC*, and *SSGC*, with results summarized in Table 8. Across both additional datasets and GNN architectures, LLM-enhanced GNNs generally exhibit higher membership leakage than the *Shallow* baseline, with varying degrees of increase across feature enhancers and GNN backbones.

B.7 Evaluation on Additional Graph Domains

To further examine whether the observed privacy risks generalize beyond the original benchmark, we extend the evaluation to four additional datasets from previously underrepresented graph domains: *WikiCS* (Mernyei & Cangea, 2020), *USA-Airports* (Ribeiro et al., 2017), *Texas* (Craven et al., 1998),

and *Wisconsin* (Craven et al., 1998). Together with the original six datasets, the expanded benchmark contains ten datasets spanning academic citation, social networks, e-commerce, Wikipedia, transportation, and university websites. As shown in Table 9, Linq-GCN consistently exhibits higher attack performance than Shallow-GCN across all 12 comparisons covering link, label, and membership inference. On average, Linq-GCN increases attack performance by 6.63, 10.14, and 4.24 percentage points for link, label, and membership inference, respectively. The increases are particularly pronounced for label inference on Texas and Wisconsin, where the gaps reach 15.74 and 13.94 percentage points. These results extend our observations to substantially different graph domains and provide further evidence that the increased empirical privacy vulnerability is not specific to the original benchmark datasets.

Table 9: Privacy attack performance (%) on four additional datasets using Shallow-GCN and Linq-GCN. Link, label, and membership inference are evaluated using `MLA`, `NIA`, and `MIA`, respectively. Results are reported as mean (standard deviation) over five runs.

Dataset	Method	Link AUC	Label Acc.	Membership AUC
WikiCS	Shallow-GCN	81.63 (0.33)	47.80 (8.53)	50.29 (0.96)
	Linq-GCN	83.76 (0.58)	52.18 (6.12)	52.85 (1.17)
USA-Airports	Shallow-GCN	65.04 (9.28)	45.31 (2.33)	50.08 (3.46)
	Linq-GCN	75.10 (1.24)	51.79 (4.55)	53.43 (3.20)
Texas	Shallow-GCN	61.04 (2.88)	47.33 (6.88)	46.65 (5.03)
	Linq-GCN	71.01 (4.77)	63.07 (6.81)	51.62 (8.99)
Wisconsin	Shallow-GCN	68.73 (5.76)	41.34 (5.51)	45.46 (2.02)
	Linq-GCN	73.07 (0.90)	55.28 (9.66)	51.53 (5.58)

B.8 Evaluation with More Recent LLM Backbones

To examine whether the observed privacy risks remain relevant with more recent language models, we further evaluate three recent LLM backbones: *Llama-3.1-8B* (Grattafiori et al., 2024), *Qwen3-Embedding-8B* (Zhang et al., 2025a), and *DeepSeek-V4-Pro* (DeepSeek-AI, 2026). These models were released between July 2024 and April 2026, substantially extending the temporal coverage of the LLMs considered in our main evaluation. We conduct the experiments on Tape-Arxiv23 using the same GCN backbone, data split, training configuration, and attack protocols as in the main experiments. All results are reported as mean $\pm$ standard deviation over five runs.

As shown in Table 10, all three recent LLM backbones exhibit higher attack performance than the *Shallow* baseline across link, label, and membership inference. The increase is particularly pronounced for label inference, where attack accuracy rises by 14.86–25.18 percentage points. For link and membership inference, the corresponding increases range from 1.07–7.62 and 4.37–6.59 percentage points, respectively. These results further show that the elevated empirical privacy vulnerability observed in our main experiments persists with more recent LLM backbones, although its magnitude varies across models and privacy threats.

Table 10: Privacy attack performance (%) of more recent LLM backbones on Tape-Arxiv23 under GCN. Link, label, and membership inference are evaluated using `MLA`, `NIA`, and `MIA`, respectively.

Representation	Release Date	Link AUC	Label Acc.	Membership AUC
Shallow	–	80.28 $\pm$ 1.74	48.08 $\pm$ 4.08	60.14 $\pm$ 2.10
Qwen3-Embedding-8B	2025-06	87.90 $\pm$ 0.49	62.94 $\pm$ 4.43	66.73 $\pm$ 1.20
Llama-3.1-8B	2024-07	84.13 $\pm$ 0.78	68.70 $\pm$ 3.69	66.56 $\pm$ 0.49
DeepSeek-V4-Pro	2026-04	81.35 $\pm$ 1.62	73.26 $\pm$ 1.01	64.51 $\pm$ 1.70

B.9 Comparison with a Vanilla Pretrained LM

Comparing only against shallow text representations may confound the effect of general semantic richness with that of LLM-specific enhancement. We therefore include *Pure DeBERTa* (He et al., 2021a) as an intermediate baseline. Pure DeBERTa directly encodes the original node text without LLM-generated explanations or prompt-based augmentation. We use Tape-Arxiv23 with the GCN backbone and keep the dataset split, training protocol, and attack settings unchanged. Results are averaged over five runs. As shown in Table 11, Pure DeBERTa exhibits higher attack performance than Shallow across all six attacks, indicating that richer pretrained representations explain part of the increased privacy vulnerability. However, LLaMA remains consistently more vulnerable than Pure DeBERTa, showing that the observed privacy gap cannot be attributed solely to the use of pretrained language representations.

Table 11: Attack performance of Shallow, Pure DeBERTa, and LLaMA on Tape-Arxiv23 with GCN.

Method	MIA	NMA	MLA	ELA	HGA	NIA
Shallow	60.1 ± 2.1	51.3 ± 0.8	80.3 ± 1.7	76.7 ± 2.5	62.8 ± 0.8	38.1 ± 4.1
Pure DeBERTa	64.3 ± 3.0	56.1 ± 2.5	80.9 ± 0.6	77.3 ± 0.8	65.0 ± 0.7	45.6 ± 4.3
LLaMA	67.6 ± 0.5	62.4 ± 1.0	85.1 ± 0.8	81.9 ± 0.6	67.9 ± 1.0	62.3 ± 2.2

B.10 Analysis of Potential Confounding Factors

B.10.1 Effect of Representation Dimensionality

LLM/LM-based representations may differ substantially in dimensionality from shallow features, potentially confounding privacy comparisons. To control for this factor, we conduct a dimension-matched experiment on Cora with the GCN backbone. For each representation, we independently fit a label-independent Truncated SVD projection using only the training nodes and project the features to 768 dimensions. The learned projection is then applied to validation and test nodes. TAPE already has 768 dimensions and is therefore retained without projection. All other training and attack settings remain unchanged. As shown in Table 12, after dimensionality matching, all six semantic enhancement methods remain more vulnerable than Shallow across the three privacy threats. On average, they exceed Shallow by 1.8, 8.5, and 1.8 percentage points under MLA, NIA, and MIA, respectively. These results suggest that representation dimensionality alone does not explain the observed empirical privacy gap.

Table 12: Attack performance (%) after matching all feature representations to 768 dimensions on Cora with GCN. MLA, NIA, and MIA represent link, label, and membership inference, respectively.

Attack	**Shallow**	**TAPE**	**KEA**	**LLaMA**	**Linq**	**SimTeg**	**E5-Large**
MLA	68.9	71.2	69.9	71.7	70.6	70.1	70.8
NIA	57.4	60.5	61.3	60.7	72.8	64.2	75.9
MIA	73.9	75.5	75.7	75.2	76.0	75.8	76.2

B.10.2 Privacy Risk under Matched Predictive Utility

Differences in predictive utility may also confound the relationship between semantic enhancement and privacy vulnerability. We therefore conduct a matched-accuracy analysis using Shallow-GCN and Linq-GCN on Reddit. We adjust the training epochs and regularization of Linq-GCN so that its node-classification accuracy closely matches that of Shallow-GCN, while keeping the dataset split, GNN architecture, and attack configurations unchanged. Results are averaged over five runs. As shown in Table 13, with closely matched predictive utility, Linq remains more vulnerable than Shallow across link, label, and membership inference, increasing attack performance by 1.60, 2.41, and 1.57 percentage points, respectively. This result indicates that the increased empirical privacy vulnerability cannot be explained solely by higher node-classification accuracy.

Table 13: Attack performance under closely matched node-classification utility on Reddit with GCN.

Method	Utility	Link	Label	Membership
Shallow	59.91 ± 0.19	60.53 ± 4.52	52.46 ± 5.44	51.15 ± 1.33
Linq	60.26 ± 0.94	62.13 ± 3.55	54.87 ± 6.48	52.72 ± 0.85

B.11 Cross-Backbone Mechanistic Analysis

To examine whether the mechanistic observations in Fig. 6 generalize beyond GCN, we repeat the same distributional analysis on Tape-Arxiv23 using SAGE and GAT, while keeping the feature representations and evaluation settings unchanged. The complete CDF results are reported in Tables 14 and 15. At a threshold of 0.6, the link-, label-, and membership-related CDF gaps increase from 0.399, 0.341, and 0.100 for Shallow to 0.604, 0.644, and 0.289 for Linq under SAGE. Under GAT, the corresponding gaps increase from 0.132, 0.116, and 0.014 to 0.403, 0.364, and 0.145. These consistent patterns indicate that the stronger separation of privacy-related signals observed under GCN also appears across different GNN backbones.

Table 14: CDF results under SAGE for the mechanistic analysis of link, label, and membership inference.

Threat	Curve	0.2	0.4	0.6	0.8	1.0
Link Inference	Shallow-Connected	0.177	0.529	0.866	0.993	1.000
	Shallow-Unconnected	0.013	0.132	0.467	0.894	1.000
	Linq-Connected	0.181	0.599	0.872	0.982	1.000
	Linq-Unconnected	0.020	0.094	0.268	0.691	1.000
Label Inference	Shallow-Intra	0.059	0.372	0.773	0.983	1.000
	Shallow-Inter	0.006	0.102	0.432	0.886	1.000
	Linq-Intra	0.137	0.507	0.833	0.980	1.000
	Linq-Inter	0.002	0.033	0.189	0.660	1.000
Membership Inference	Shallow-Member	0.177	0.441	0.643	0.802	1.000
	Shallow-Non-member	0.263	0.553	0.743	0.872	1.000
	Linq-Member	0.004	0.028	0.070	0.145	1.000
	Linq-Non-member	0.045	0.190	0.359	0.541	1.000

Table 15: CDF results under GAT for the mechanistic analysis of link, label, and membership inference.

Threat	Curve	0.2	0.4	0.6	0.8	1.0
Link Inference	Shallow-Connected	0.950	0.992	0.998	1.000	1.000
	Shallow-Unconnected	0.107	0.539	0.866	0.988	1.000
	Linq-Connected	0.427	0.931	0.997	1.000	1.000
	Linq-Unconnected	0.003	0.138	0.594	0.952	1.000
Label Inference	Shallow-Intra	0.351	0.869	0.988	1.000	1.000
	Shallow-Inter	0.091	0.522	0.872	0.991	1.000
	Linq-Intra	0.030	0.579	0.954	0.999	1.000
	Linq-Inter	0.002	0.107	0.590	0.954	1.000
Membership Inference	Shallow-Member	0.667	0.871	0.953	0.991	1.000
	Shallow-Non-member	0.729	0.901	0.967	0.994	1.000
	Linq-Member	0.001	0.011	0.048	0.105	1.000
	Linq-Non-member	0.008	0.074	0.193	0.327	1.000

B.12 Private GNN Training with GAP

To complement the input-level LDP defenses evaluated in Sec. 5.6, we further examine private GNN training using GAP (Sajadmanesh et al., 2023). Experiments are conducted on Cora with Linq-enhanced features and a GCN backbone, while keeping the dataset split, training configuration, and attack settings unchanged. We evaluate three representative attacks, namely MLA, NIA, and MIA, corresponding to link, label, and membership inference, respectively.

B.12.1 Effectiveness of GAP

Table 16 reports model utility and attack performance under different GAP privacy budgets. Compared with non-private training, GAP consistently reduces the effectiveness of all three privacy attacks, while stronger privacy protection incurs greater utility degradation. As shown in Table 16, smaller privacy budgets provide stronger protection but lead to lower predictive utility. For example, at $\epsilon_g = 2.0$, MLA, NIA, and MIA decrease by 11.5, 15.9, and 23.2 percentage points, respectively, compared with non-private training, while utility decreases by 10.0 points. These results show that private GNN training can substantially mitigate the evaluated privacy attacks, albeit with a noticeable utility cost.

Table 16: Effectiveness of GAP on Cora with Linq-GCN under different privacy budgets. All values are percentages.

Setting	ϵ_g	**Utility**	MLA	NIA	MIA
Non-private	∞	84.8	71.3	75.0	76.8
GAP	1.0	69.4	55.2	54.7	47.9
GAP	2.0	74.8	59.8	59.1	53.6
GAP	4.0	77.9	64.7	64.1	59.8

B.12.2 Utility under Matched Privacy Budgets

We further compare Shallow-GCN and Linq-GCN under identical GAP privacy budgets to examine whether LLM-enhanced representations exhibit different utility behavior after privatization. The same dataset split, GCN backbone, training configuration, and GAP mechanism are used for both models. The results are reported in Table 17. As shown in Table 17, Linq-GCN achieves slightly lower utility than Shallow-GCN under all three matched privacy budgets, with gaps of 1.2, 0.8, and 0.7 percentage points at $\epsilon_g = 1.0$, 2.0, and 4.0, respectively. Under the evaluated setting, these results suggest that the utility advantage of semantic enhancement becomes more difficult to preserve when formal privacy protection is imposed.

Table 17: Utility accuracy (%) of Shallow-GCN and Linq-GCN under matched GAP privacy budgets on Cora.

Method	$\epsilon_g = 1.0$	$\epsilon_g = 2.0$	$\epsilon_g = 4.0$
Shallow-GCN	70.6	75.6	78.6
Linq-GCN	69.4	74.8	77.9

B.13 Defense Generalization across GNN Backbones

To examine whether the defense observations are specific to GCN, we further extend the DP evaluation to *GAT* and *SAGE*. Following the same feature-enhancement and attack settings as the main defense evaluation, we evaluate embedding-, link-, and label-level DP under privacy budgets of 0.1, 0.5, and 1.0. For embedding- and link-level DP, attack performance is measured by AUC (%), while label-level attack performance and model utility are measured by accuracy (%). The complete results are reported in Table 18. As shown in Table 18, the same overall privacy–utility trend is observed for both GAT and SAGE across all three protection levels. Smaller privacy budgets consistently suppress attack performance, but simultaneously reduce node classification utility. These results are consistent

with the observations under GCN, indicating that the effectiveness and utility costs of the evaluated DP defenses are not specific to a single GNN backbone.

Table 18: Defense performance across GAT and SAGE under different privacy budgets. "Attack" denotes the corresponding privacy attack performance, and "Utility" denotes node classification accuracy (%).

DP Level	Backbone / Metric	0.1	0.5	1.0
Embedding	GAT / Attack AUC	40.3	43.8	47.4
	GAT / Utility	42.8	45.7	49.3
	SAGE / Attack AUC	41.1	42.7	47.9
	SAGE / Utility	41.9	47.2	48.6
Link	GAT / Attack AUC	40.4	43.0	46.7
	GAT / Utility	41.3	43.5	45.8
	SAGE / Attack AUC	39.2	44.1	45.5
	SAGE / Utility	42.6	43.0	47.1
Label	GAT / Attack Acc.	28.6	30.4	34.7
	GAT / Utility	34.4	36.8	38.6
	SAGE / Attack Acc.	27.4	31.6	33.8
	SAGE / Utility	33.6	37.9	39.0

C Related Work

In this section, we introduce more related work on graph neural networks (▷ Sec. C.1), LLM-enhanced GNNs (▷ Sec. C.2), and privacy attacks on GNNs (▷ Sec. C.3).

C.1 Graph Neural Networks

Graph neural networks have emerged as a powerful paradigm for learning on graph-structured data, with early works such as GCN (Kipf & Welling, 2017) and SAGE (Hamilton et al., 2017) establishing the message passing framework as a *de facto* standard. Subsequent efforts have focused on improving expressiveness and scalability, introducing mechanisms such as attention-based aggregation (Velickovic et al., 2018) and deeper architectures (Klicpera et al., 2019; Wu et al., 2019; Zhu & Koniusz, 2021). GNNs have been successfully applied to a broad range of tasks, including node classification, link prediction, and graph-level reasoning, achieving state-of-the-art results across numerous benchmarks (Dwivedi et al., 2023; Maekawa et al., 2022; Li et al., 2023). In this work, we adopt representative GNN architectures as downstream backbones to evaluate how different graph learning models interact with LLM-enhanced node features in terms of privacy vulnerability.

C.2 LLM-Enhanced GNNs

Graph neural networks have achieved strong performance on graph-structured data by aggregating neighborhood information (Kipf & Welling, 2017; Velickovic et al., 2018; Hamilton et al., 2017), yet their ability to handle rich textual node attributes remains limited. Recent studies have explored integrating LLMs with GNNs to address this limitation, giving rise to *LLM-enhanced GNNs* (He et al., 2024; Chen et al., 2024; Zhu et al., 2024; Chien et al., 2022), which significantly improve node representations and downstream task performance. Existing methods fall into two categories: *explanation-based methods*, which leverage the reasoning and knowledge capabilities of LLMs to generate high-level semantic descriptions (*e.g.*, explanations and summaries) that are subsequently encoded by a language model (He et al., 2024; Wang et al., 2022); and *embedding-based methods*, which bypass explicit text generation and instead directly encode textual attributes into dense embeddings via LLMs or fine-tuned language models (Duan et al., 2023; Wang et al., 2022). The latter can be further divided into methods that directly use LLM APIs for embedding inference and those that fine-tune smaller pretrained models for task-specific adaptation. While these approaches have demonstrated significant utility gains across various graph benchmarks, the privacy implications of incorporating semantically enriched representations into graph learning remain largely unexplored, motivating our systematic investigation.

C.3 Privacy Attacks on GNNs

Privacy risks in graph learning (Zhang et al., 2024) have been studied under three main threat scenarios. Membership inference attacks (Olatunji et al., 2021; Guan et al., 2025; He et al., 2021c) aim to determine whether a given node was included in training by exploiting behavioral discrepancies between member and non-member samples. Link inference attacks (Wu et al., 2022a; He et al., 2021b) seek to recover sensitive graph structure from learned node embeddings. Label inference attacks (Meng et al., 2023) attempt to infer private node labels by leveraging the class-discriminative structure of the representation space. While prior work has examined these threats in conventional GNNs with shallow features, their interplay with semantically enriched LLM-based representations remains an open and important question that this work aims to address.

D Limitation & Broader Impact

D.1 Limitation

This work provides a comprehensive and large-scale empirical audit of privacy risks in LLM-enhanced GNNs, spanning ten text-attributed graph datasets, six representative inference attacks, diverse feature enhancement methods, multiple GNN backbones, and complementary defense settings. Despite this broad evaluation, the current study primarily focuses on static graphs and commonly used GNN architectures. Future work could extend the evaluation to dynamic graphs, broader attack settings, and privacy-preserving mechanisms tailored to LLM-enhanced graph learning.

D.2 Broader Impact

This work provides a systematic investigation of privacy risks in LLM-enhanced GNNs, an emerging and increasingly important paradigm in graph learning. Our results show that LLM-enhanced GNNs generally exhibit greater empirical vulnerability to membership, link, and label inference under the evaluated threat models, highlighting important privacy considerations for practical deployment. We further investigate differential privacy (DP) as a mitigation strategy and analyze the associated privacy-utility trade-off. We hope this work encourages researchers and practitioners to integrate privacy considerations into the development lifecycle of LLM-enhanced graph learning systems, and motivates future research on designing more effective and utility-preserving privacy protection mechanisms for semantic-enhanced graph learning.