DP-GPL: Differentially Private Graph Prompt Learning

Jing Xu, Franziska Boenisch, Iyiola Emmanuel Olatunji, and Adam Dziedzic

CISPA Helmholtz Center for Information Security, Germany

Abstract. Graph Neural Networks (GNNs) have shown remarkable performance in various applications. Recently, *graph prompt learning* has emerged as a powerful GNN training paradigm, inspired by advances in language and vision foundation models. Here, a GNN is pre-trained on public data and then adapted to sensitive tasks using lightweight graph prompts. However, using prompts from sensitive data poses privacy risks. In this work, we are the first to investigate these practical risks in graph prompts by instantiating a membership inference attack that reveals significant privacy leakage. We also find that the standard privacy method, DP-SGD, fails to provide practical privacy-utility trade-offs in graph prompt learning, likely due to the small number of sensitive data points used to learn the prompts. As a solution, we propose `DP-GPL` for differentially private graph prompt learning based on the PATE framework, that generates a graph prompt with differential privacy guarantees. Our evaluation across various graph prompt learning methods, GNN architectures, and pre-training strategies demonstrates that our algorithm achieves high utility at strong privacy, effectively mitigating privacy concerns while preserving the powerful capabilities of prompted GNNs as powerful foundation models in the graph domain.

Keywords: Graph Prompt Learning · Membership Inference Attack · Differential Privacy.

1 Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for learning representations of graph-structured data and have shown significant advancements across various applications, such as drug design [2, 31], anomaly detection [40, 42], and social network analysis [5]. Recently, *graph prompt learning* [41, 50, 38, 12, 36, 37] has emerged as a promising GNN training paradigm. Graph prompt learning first pre-trains a GNN model on general public graph data and then tunes a graph prompt [38, 18, 13] or tokens [12, 36, 22] on some sensitive downstream data. By reformulating the downstream task into the pretext task used in pre-training, it then enables predictions for the downstream task.

The fact that graph prompts are tuned on sensitive downstream data can raise significant privacy concerns. In fact, in the language and vision domains, it has been shown that private information from downstream data can leak through

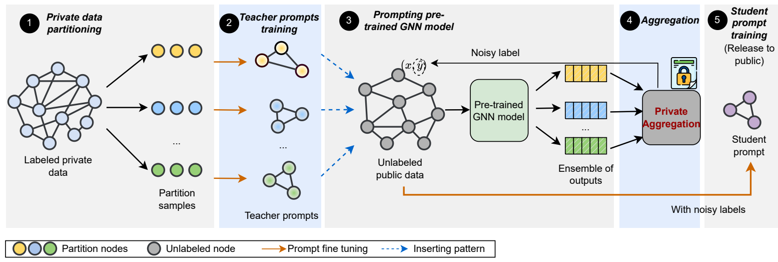


Fig. 1: Framework of DP-GPL. ❶ We partition the labeled private data into disjoint groups randomly. ❷ An ensemble of teacher prompts is trained on the disjoint private data groups. ❸ Given an unlabeled public data sample, by querying the pre-trained GNN model, each teacher prompt votes with the most confident class label. ❹ The teacher prompts' votes are privately aggregated, *i.e.*,, a noisy argmax over vote counts is returned as the final noisy label for the public data sample. ❺ A student prompt is trained with the labeled public data and can be publicly released.

predictions of prompted models [9, 45]. To the best of our knowledge, no such insights exist for the graph domain, and no prior work has explored the privacy risks of graph prompt learning.

In this work, we set out to close this gap. We first assess the privacy risks of graph prompts by adapting a state-of-the-art membership inference attack [35, 4] to graph prompt learning and measuring the empirical leakage. Our evaluation demonstrates significant privacy risks for the downstream data when used to tune graph prompts. For example, we show that the membership inference attack can achieve an AUC score as high as 0.91 on the PubMed dataset. We also investigate the relationship between the number of data points used to tune the prompt and the attack success and find that with less data, the privacy risk grows, posing a significant risk to standard graph prompt learning that usually relies on a small number of data points [37].

As a naive solution to mitigate this privacy risk, we first turn to the Differential Privacy-Stochastic Gradient Descent (DP-SGD) algorithm [1]—a gold standard in privacy-preserving machine learning. However, we find that this approach significantly degrades the downstream performance due to the limited amount of data used to tune graph prompts. For instance, with a privacy budget as high as $\varepsilon = 64$, the accuracy on the Cora dataset downstream drops from 48.70% to 18.47%, *i.e.,* close to random guessing.

As a solution for practical privacy-preserving graph prompt learning, we propose *DP-GPL*. DP-GPL follows the general framework of the private aggregation of teacher ensembles (PATE) [29, 30], but instead of training a student model with differential privacy guarantees, it trains a student prompt [8]. We thoroughly evaluate our DP-GPL in terms of privacy guarantees and privacy-utility trade-offs. Over various graph prompt learning methods, GNN architectures, and pre-training strategies, we find that our algorithm can achieve high utility at strong privacy privacy guarantees—thereby, implementing the first practical approach to private graph prompt learning.

In summary, we make the following contributions:

– We are the first to show that private information can leak from graph prompts, in particular when the prompts are tuned over a small number of data points.
– We show that naively integrating the DP-SGD algorithms into graph prompt learning yields impractical privacy-utility trade-offs.
– As a solution, we propose `DP-GPL`, an algorithm based on the PATE framework to implement differential privacy guarantees into graph prompt learning.
– We perform a thorough evaluation on multiple state-of-the-art graph prompt learning methods, graph datasets, GNN models, and pre-training strategies and highlight that our method can achieve both high utility and strong privacy protections over various setups.

2 Background and Related Work

2.1 Graph Prompt Learning

GNNs achieve strong performance on numerous applications [39, 42, 5]. Therefore, they rely on various effective architectures, such as Graph Convolutional Network (GCN) [20], Graph Attention Network (GAT) [43], and Graph Transformer [34]—usually trained in a *supervised* manner. To make graph learning more adaptive, many graph pre-training approaches have been proposed [44, 16, 36, 46] that first learn some general knowledge for the graph model with easily accessible data, and then fine-tune the model on new tasks. This is often referred to as *"pre-train & fine-tune"* paradigm. However, the large diversity between graph tasks with node level, edge level, and graph level may cause a "negative transfer" results where the knowledge learned during the pre-training phase hurts performance when fine-tuning on a specific downstream task, rather than improving it [38]. As a solution, graph prompt learning was proposed. The goal of graph prompt learning is to learn transformation operations for graphs to reformulate the downstream task to the pre-training task. Given an input graph $\mathcal{G}$, we can formulate the graph prompt learning as follows:

$$\mathcal{G}^* : (X^*, A_{inner}, A_{insert}) = \mathcal{P}(X, A) \tag{1}$$

where $\mathcal{G}^*$ is a prompted graph, which includes learnable components in its feature matrix, *i.e.,* $X^* \in \mathbb{R}^{K \times d}$ and adjacency matrix, *i.e.,* A_{inner} and A_{insert}. $\mathcal{P}$ is a graph prompt learning module that learns the representations of K prompt tokens, *i.e.,* X^*, token structures, *i.e.,* A_{inner} and inserting patterns, *i.e.,* A_{insert}, which indicates the connection between the prompt tokens and the nodes in the original graph. We can learn a graph prompt learning module $\mathcal{P}$ applied to the original graph to imitate any graph-level transformation. While Equation (1) shows graph-level transformation, our adaption of graph prompt is in node-level, *i.e.,* the graph prompt is learned only based on the selected nodes' features without the adjacency matrix A of the original graph $\mathcal{G}$. In addition, the learned graph prompt is adapted to individual nodes, *i.e.,* $\mathcal{P}(x)$ where x is an individual node.

For instance, Graph Pre-training and Prompt Tuning (GPPT) [36] applies prompt-based tuning methods to models pre-trained by edge prediction. It introduces virtual class-prototype nodes/graphs with learnable links into the original graph, making the adaptation process more akin to edge prediction. [12] proposed a universal prompt-based tuning method, called Graph Prompt Feature (GPF), which can be applied under any pre-training strategy. GPF adds a shared learnable vector to all node features in the graph while its variant GPF-plus incorporates different prompted features for different nodes in the graph. [38] proposed All-in-one, a graph prompt that unifies the prompt format in the language area and graph area with the prompt token, token structure, and inserting pattern. They reformulate the downstream problems to the graph-level task to further narrow the gap between various graph tasks and pre-training strategies. Graph prompt learning has superior performance compared to traditional fine-tuning methods and is especially effective in few-shot settings, *i.e.,* when only a small number of data points are sampled to tune the prompt. While graph prompt learning benefits various graph applications, in this work, we focus on node classification tasks and three state-of-the-art graph prompt learning methods, namely GPPT, All-in-one, and GPF-plus.

Although some prior work explores backdoor attacks in graph prompt learning, which utilize prompts to insert backdoor triggers into the GNN model [23] to impact output integrity, to the best of our knowledge, there is no prior work on assessing and mitigating the privacy risks in graph prompt learning.

2.2 Differential Privacy

Differential privacy (DP). DP [10] is a mathematical framework that provides privacy guarantees for randomized mechanisms $\mathcal{M} : I \to S$. Therefore, it upper-bounds the probability that $\mathcal{M}$, when executed on two neighboring datasets D, D', *i.e.,* dataset that differ in only one data point, output a different result by formalizing that $\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in S] + \delta$. The privacy parameter ε specifies by how much the output is allowed to differ, and δ is the probability of failure to meet that guarantee. There are two main algorithms to implement DP guarantees for traditional machine learning. The **differentially private stochastic gradient descent** algorithm (DP-SGD) [1] extends standard stochastic gradient descent with two additional operations, first, gradient clipping that limits the impact of each individual training data point (often called *"sensitivity"*) on the model update, and then the addition of calibrated amounts of stochastic noise to provide formal privacy guarantees. The second **private aggregation of teacher ensembles** algorithm (PATE) [29, 30] trains an ensemble of *teacher* models on disjoint subsets of the private data. Then, through a noisy labeling process, the ensemble privately transfers its knowledge to an unlabeled public dataset. Finally, a separate *student* model is trained on this labeled public dataset for release.

DP for Graphs. As the classical DP guarantee makes no assumptions about potential correlations between data points, there are existing works that extend DP on graph data [26, 33, 19, 27, 32, 47]. There are three variants of DP on graph data: node-level DP, edge-level DP, and graph-level DP, depending on what the data owner requires to protect. Specifically, node-level DP aims to protect the privacy of individual nodes in the graph data, including its attributes and associated edges [33, 19, 7, 27]. Edge-level DP aims to protect the relationships between nodes, which can be applied to social network graphs [14] or location graphs [48], where the edges contain sensitive information, but the data represented in the nodes of the graph are assumed to be non-sensitive. Graph-level DP aims to protect the entire graph data, including the structure of the graph, node attributes, and edge relationships [25]. However, graph-level DP has not been thoroughly investigated in the literature [26]. In this work, we focus on *node-level DP* as we aim to protect the privacy of individual nodes in the graph data. Different from the existing node-level DP guarantees [33, 19, 27, 32, 47], which often results in large ϵ values, limiting their practical utility, we aim to achieve meaningful privacy guarantees for graph prompt learning with small and manageable ϵ values ($\epsilon <= 2$).

2.3 Private Prompt Learning in the Vision and Language Domain

In the vision domain, [21] leverage the PATE algorithm for private prompt tuning to vision encoders. Therefore, they have to tune a prompt and train an additional label mapping for each teacher. In contrast, our method instantiates different teachers only through graph input prompts. In the language domain, multiple approaches have been proposed to privatize prompts. [6] rely on named entity recognition to identify and hide private information in text prompts. This approach is not easily transferable to the graph domain and additionally does not yield formal privacy guarantees. The DP-OPT [15] framework relies on a local large language model (LLM) to derive a discrete, *i.e.,* text, prompt with DP, and then transfers this prompt to a central LLM. The framework is tightly coupled to the language domain and derives plain language prompt templates that are not applicable to GNNs. [28] rely on a PATE-style teacher ensemble implemented through different prompts, and generate noisy output predictions for the LLM. Yet, due to the absence of a student model in their framework, each query to the ensemble consumes additional privacy budget, making the approach impractical. [8] solve this limitation by generating a student prompt from the teacher ensemble, similar to our work.

3 Privacy Risks in Graph Prompt Learning

In this work, we explore the privacy risk for the sensitive downstream data in graph prompt learning by instantiating a MIA[4, 35]. While prior work on instantiating MIAs against natural language prompts relies on a simple threshold-based attack [9], we adapt and implement the more powerful state-of-the-art Likelihood Ratio Attack (LiRA) [4].

We use this attack to assess whether a given data point was used to train a given target prompt. Formally, in our MIA, we consider that the goal of the adversary is to infer whether a given private data sample $v = (x_p, y_p)$ is in the training dataset of the target prompt $\mathcal{P}_{target}$. We assume that the adversary holds n candidate nodes $(x_1, x_2, \ldots, x_n)$ including their corresponding labels $(y_1, y_2, \ldots, y_n)$ and queries the candidates nodes with prepended target prompt to the pre-trained GNN model.

Algorithm 1 Likelihood Ratio Attack on Graph Prompt Learning. Instead of conducting MIA against the target model in the standard LiRA algorithm, we conduct MIA against the target prompt in graph prompt learning. We highlight these differences in blue.

Require: Target prompt $\mathcal{P}_{target}$, Pre-trained GNN model Φ, A given data sample (x_p, y_p), data distribution $\mathbb{D}$, Logit scaling $f(p) = log(\frac{p}{1-p})$

confs$_{in}$ = {}, confs$_{out}$ = {}
for $i \leftarrow 1$ to K times **do**
  /* Sample a shadow dataset */
  $D_{attack} \xleftarrow{\$} \mathbb{D}$
  /* Train IN graph prompt */
  $\mathcal{P}_{in} \leftarrow \mathcal{T}(D_{attack} \cup (x_p, y_p))$
  confs$_{in}$ ← confs$_{in}$ ∪ $\{f(\Phi(\mathcal{P}_{in}(x_p))_{y_p})\}$
  /* Train OUT graph prompt */
  $\mathcal{P}_{out} \leftarrow \mathcal{T}(D_{attack} \setminus (x_p, y_p))$
  confs$_{out}$ ← confs$_{out}$ ∪ $\{f(\Phi(\mathcal{P}_{out}(x_p))_{y_p})\}$
end for
$\mu_{in} \leftarrow$ `mean` (confs$_{in}$), $\mu_{out} \leftarrow$ `mean` (confs$_{out}$)
$\sigma^2_{in} \leftarrow$ `var`(confs$_{in}$), $\sigma^2_{out} \leftarrow$ `var`(confs$_{out}$)
/* Query with target graph prompt */
conf$_{obs}$ = $f(\Phi(\mathcal{P}_{target}(x_p))_{y_p})$

Ensure: $\Lambda = \frac{p(conf_{obs} | \mathcal{N}(\mu_{in}, \sigma^2_{in}))}{p(conf_{obs} | \mathcal{N}(\mu_{out}, \sigma^2_{out}))}$

The pre-trained GNN model then outputs the probability vectors $(p_1, p_2, \ldots, p_n)$. Following [4], we analyze the model's output probability at the correct target class label of every candidate node x_i, *i.e.,* p_{i,y_i}. The intuition of this MIA is that the output probability at the correct class y_i will be significantly higher for members that were used in training $\mathcal{P}_{target}$ than non-members. The detail of our adaptation of the LiRA attack to the graph prompt learning setup is presented in Algorithm 1.

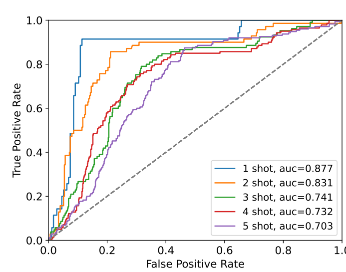


Fig. 2: **AUC-ROC curve of our MIA on Cora dataset with different number of shots, *i.e.,* 1-5 shots.** With fewer shots, MIA success rises significantly.

MIA Experimental Setup. We conduct MIA against graph prompt learning on three downstream datasets, *i.e.,* Cora, CiteSeer, and PubMed with GNN models pre-trained on the ogbn-arxiv dataset.[1] To evaluate our MIAs under different numbers of data points used

[1] Details of these datasets are presented in Section 5.1.

to tune the graph prompt, we analyze MIA in 1-5 shot settings. Following the experimental setup from MIAs against natural language prompts [9], for each experiment, we consider the k (*i.e.,* k=1-5) data points used in training the target prompt as members and $50 * k$ other randomly selected data points from the testing dataset as non-members. We repeat the MIA 100 times and report the average attack success.

MIA Results. In Figure 2, we present the AUC-ROC curve of our MIA on the Cora dataset and the GAT model. The results for other datasets and models are presented in Appendix A.4 and show a similar trend. Our results highlight that the privacy risk increases with fewer shots used to train the prompt, *e.g.,* with 5 shots we have an AUC score of 0.703, while with 1 shot, the AUC score increases to 0.877. We hypothesize that this is due to the fact that with fewer shots, the target prompt is more likely to overfit the prompt data, leading to a higher membership inference risk. Yet, even with more shots, we observe significantly higher MIA success than the random guessing (0.5), *e.g.,* see Figure 3 with 5-shots over various setups where the average AUC score is consistently between 0.7-0.9. Hence, our results demonstrate that the private data used in training a graph prompt can be subject to substantial privacy risk. This motivates the urgent need for privacy-preserving graph prompt learning methods.

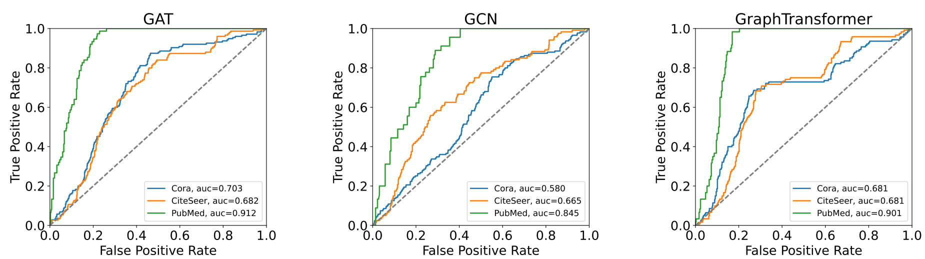


Fig. 3: **AUC-ROC curve of our MIA (with 5 shots).** Generally, there is a high MIA risk in terms of AUC score of between 0.7-0.9.

4 Towards Privacy Preserving Graph Prompts

The standard approach for privacy-preserving machine learning is based on the DP-SGD algorithm [1]. The DP-SGD algorithm can be applied in gradient-based learning approaches to limit the impact of individual training data points on the final model and add calibrated noise to implement the privacy guarantees. We explore this naive way of implementing privacy guarantees into graph prompt learning and show that it fails to yield reasonable utility even at low privacy regimes, *i.e.,* with very high ε's. Motivated by this insight, we propose a non-gradient based algorithm for private graph prompt learning based on the PATE framework.

4.1 Naive Implementations of Privacy in Graph Prompt Learning Fail

As a naive solution to yield private graph prompt learning, we rely on the DP-SGD algorithm. Therefore, we keep the GNN frozen, calculate the gradients only with respect to the graph prompts, clip and noise them according to the desired privacy protection, and update the prompt iteratively to minimize the loss on the downstream task. Our evaluation of this naive approach in Table 8 in Appendix A.4 highlights that DP-SGD yields inadequate privacy-utility trade-offs for private graph prompt learning. While our results show the general trend that with increasing privacy budgets, the performance of the downstream task increases, DP-SGD still significantly degrades the downstream task performance even at high privacy budgets. For instance, with a privacy budget as high as $\varepsilon = 64$ in the 5-shot setting, the accuracy of the downstream task on the Cora dataset still drops from 48.70% to 18.47%, which is close to random guessing.

4.2 Differentially Private Graph Prompt Learning Framework

Motivated by the failure of the naive DP-SGD approach, we propose a non-gradient based DP graph prompt learning framework, `DP-GPL`. We detail the general workflow of `DP-GPL` in Figure 1.

Following PATE [29, 30], our algorithms contain the broader stages of training the teacher models, performing a private knowledge transfer, and obtaining the student. In contrast to standard PATE, we do not train teachers from scratch, but using the same frozen pre-trained GNN, we tune teacher prompts. Additionally, our student is again not a trained model like in PATE, but a prompt tuned on the public data labeled during the knowledge transfer. We detail the building blocks of our `DP-GPL` below:

Private Data Partition and Teacher Prompt Tuning. In `DP-GPL`, we first partition the labeled private data into disjoint groups randomly and assign each partition to each teacher. Then, we tune the teacher prompts according to the data points that were assigned to them. The teacher *prompt* tuning differs from PATE which trains teacher *models* from scratch.

Public Querying. To label the public data based on the teacher ensemble, we infer it through the prompted GNN. Therefore, for each teacher, we need to first insert the teacher prompt into the public data. How this insertion is done differs among different graph prompt learning methods. For example, in the GPF-plus method, we insert the teacher prompt into the node features of the public data samples, while in the All-in-one method, the teacher prompt is inserted into the public data as an extra subgraph. Then, we query the pre-trained GNN model once per teacher. For each teacher, we take as a vote the class label with the highest confidence.

Noisy Teacher Vote Aggregation. In `DP-GPL`, we aggregate the teachers' votes with a majority voting mechanism akin to PATE. Specifically, for a query $\mathcal{Q}$ from the downstream task and classes 1 to C, let $y_i(\mathcal{Q}) \in [1, C]$ denote the pre-trained GNN model's prediction for i-th teacher prompt, and $c_m(\mathcal{Q})$

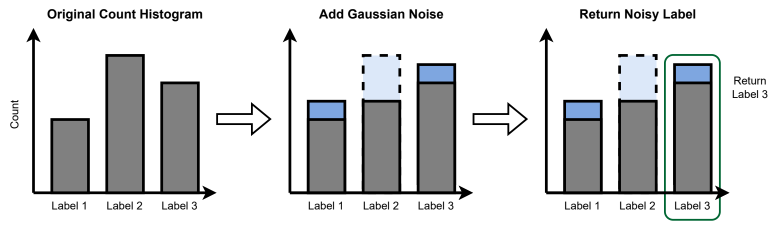


Fig. 4: **Private Aggregation.** An overview of the aggregation stage in `DP-GPL`. We first count the output labels and show them in a histogram. Then, we add Gaussian noise to the votes and return the vote with the highest noisy count as the returned label for the public data sample.

denote the vote count for class m, *i.e.*, $c_m(\mathcal{Q}) = \sum_i^N (y_i(\mathcal{Q}) = m)$. Then, we add independent Gaussian noise to the count for each class, following the Confident GNMax algorithm [30], and return the label with the highest noisy count for the query.

Student Prompt Training. Instead of training a student *model*, like in the original PATE, we use the labeled public data from the aggregation stage to train a *student graph prompt*. This prompt can be released to the public while protecting the private data used to train the teacher prompts.

4.3 Privacy Analysis

As the training nodes for different teacher prompts are independent and do not have connecting edges, the privacy analysis of our methods follows that in the original PATE algorithm [30]. We analyze the privacy analysis of `DP-GPL` below.

The privacy analysis of `DP-GPL` follows the standard privacy analysis of the GNMax algorithm, see [30], Section 4.1. Let $f(x)$ denote the histogram obtained by the teacher votes. We use the Gaussian mechanism [11] to obtain a noisy histogram $f'(x)$ as $f'(x) = f(x) + \mathcal{N}(0, \sigma^2)$. We denote by Δ_f the sensitivity of f.[2] The Gaussian mechanism then yields the following data independent bound for PATE [24]:

$$(\alpha, \Delta_f^2 \cdot \alpha/2\sigma^2)\text{-Rényi-DP}. \tag{2}$$

Using standard conversion [24], we can convert this bound back to (ε, δ)-DP bounds.

5 Empirical Evaluation

5.1 General Experimental Setup

Datasets. We use ogbn-arxiv [17], which is a large-scale graph dataset, as the pre-training dataset. For the downstream tasks, we use Cora [49], CiteSeer [49],

[2] Given that each teacher can contribute 1 vote, $\Delta_f = 1$ in `DP-GPL`.

and PubMed [49]. Since the pre-trained dataset (*i.e.,* ogbn-arxiv) and downstream dataset (*i.e.,* Cora, CiteSeer, and PubMed) have various input feature dimensions, we here use SVD (Singular Value Decomposition) to unify input features from all dimensions as 100 dimensions, following the process in [38]. We provide more details about these datasets in Appendix A.1. For each dataset, We randomly select 50% of the nodes as the private data and the remaining 50% as the public data. Within the public data, we randomly select 50 nodes as the query nodes and the remaining nodes as the testing data.

Models. We use three widely-used GNN models, *i.e.,* **GCN** [20], **GAT** [43], and **Graph Transformer** (GT) [34] as the backbone for both "pre-train & fine-tune" and graph prompt learning paradigms. The default hyperparameters used for pre-training GNN models are presented in Table 5. For pre-training strategies, we select four mostly used methods covering node-level, edge-level, and graph-level strategies, *i.e.,* DGI [44], GraphMAE [16], EdgePreGPPT [36], and SimGRACE [46].

Graph Prompt Learning Methods. Current popular graph prompt learning methods can be classified into two types, 'Prompt as graph' and 'Prompt as token' [50]. For 'Prompt as graph' type, we select **All-in-one** [38], and for 'Prompt as token' type, we use **GPPT** [36], and **GPF-plus** [12]. These graph prompt methods are all state-of-the-art. Also, we focus on the 5-shot graph prompt learning setting as it has high performance on downstream tasks (as shown in Table 7 in Appendix A.4) and also high MIA risk (as shown in Figure 3).

Privacy Parameters and Accounting. We set the privacy parameters for `DP-GPL` according to Table 6 in Appendix A.2. To empirically account for the per-teacher privacy loss during our experiments, we build on the code-based from [3].

`DP-GPL`**.** We use an ensemble of 200 teacher prompts, and each teacher prompt is trained with disjoint 5 shots of data from the private downstream task. For query dataset, we select 50 public samples from the downstream distribution. `DP-GPL` is implemented to immediately stop querying once a teacher has reached their privacy limit, which we set to $\varepsilon = 2$. We repeat each experiment three times and report the average and standard deviation of the public student prompt's accuracy on the testing dataset.

Baselines. We compare against three baselines. (1) *Lower Bound (LB):* ($\varepsilon = 0$). Given a pre-trained GNN model, we directly evaluate its performance on the downstream test data. (2) *Ensemble Accuracy (Ens. Acc.):*($\varepsilon = \infty$). We use the histogram of the private teacher ensemble votes and return the clean argmax. (3) *Upper Bound (UB):* ($\varepsilon = \infty$). *i.e.,* we select the teacher prompt which has the best testing accuracy.

Table 1: **Performance comparison between our DP-GPL and three baselines on three downstream datasets. (DGI, All-in-one, $\delta = 1.5 \times 10^{-4}$). LB – Lower Bound, UB – Upper Bound.** DP-GPL performs significantly better than the lower bound in all setups. Generally, there is a more than 30% improvement for DP-GPL over the lower bound.

		LB	Ens. Acc.	UB	Our DP-GPL	
	Private	$\varepsilon = 0$	$\varepsilon = \infty$	$\varepsilon = \infty$	ε	Test Acc
GAT	Cora	43.92	67.09	67.12	0.2226	57.96 ± 2.12
	CiteSeer	37.51	73.44	74.75	0.2047	73.49 ± 2.04
	PubMed	32.86	71.48	71.72	0.2383	66.07 ± 1.78
GCN	Cora	49.10	62.35	64.04	0.2025	56.22 ± 2.00
	CiteSeer	40.51	62.95	64.63	0.2001	59.41 ± 1.97
	PubMed	29.95	69.09	70.13	0.2386	62.70 ± 2.10
GT	Cora	21.80	55.36	56.77	0.2276	54.53 ± 1.97
	CiteSeer	27.56	51.75	53.51	0.3627	43.88 ± 2.13
	PubMed	39.23	70.63	72.95	0.2084	63.93 ± 2.15

Table 3: **Performance comparison between our DP-GPL and three baselines on three downstream datasets. (DGI, GPPT, $\delta = 1.5 \times 10^{-4}$).** LB – Lower Bound, UB – Upper Bound.

		LB	Ens. Acc.	UB	our DP-GPL	
	Private	$\varepsilon = 0$	$\varepsilon = \infty$	$\varepsilon = \infty$	ε	Test Acc
GAT	Cora	43.92	51.73	56.39	0.7777	46.90 ± 1.24
	CiteSeer	37.51	48.55	54.29	0.4790	42.65 ± 1.26
	PubMed	32.86	63.97	68.25	0.2874	59.55 ± 0.88
GCN	Cora	49.10	59.23	64.16	0.4980	54.15 ± 2.02
	CiteSeer	40.51	56.41	60.60	0.3728	52.09 ± 1.19
	PubMed	29.95	68.41	73.41	0.2601	63.28 ± 4.75
GT	Cora	21.80	56.84	58.74	0.6964	54.78 ± 3.15
	CiteSeer	27.56	48.28	49.76	0.5904	46.63 ± 2.86
	PubMed	39.23	66.52	69.46	0.3846	63.38 ± 2.11

5.2 Results

We present the results of our DP-GPL, and of the three baselines on different GNN models and downstream datasets in Table 1, Table 2 and Table 3, taking the DGI pertaining strategy as the examples. The results for other setups in Appendix A.4 show the same trends. We first observe that both our proposed algorithms significantly improve over the lower bound ($\varepsilon = 0$) baseline, highlighting their

Table 2: **Performance comparison between our `DP-GPL` and three baselines on three downstream datasets. (DGI, GPF-plus, $\delta = 1.5 \times 10^{-4}$).** LB – Lower Bound, UB – Upper Bound.

		LB	Ens. Acc.	UB	our `DP-GPL`	
	Private	$\varepsilon = 0$	$\varepsilon = \infty$	$\varepsilon = \infty$	ε	Test Acc
GAT	Cora	43.92	59.14	60.13	0.9186	58.10 ± 1.63
	CiteSeer	37.51	69.24	70.38	0.4917	68.11 ± 1.39
	PubMed	32.86	79.07	79.22	0.3150	78.85 ± 1.40
GCN	Cora	49.10	71.33	77.87	0.4268	64.64 ± 0.73
	CiteSeer	40.51	82.70	85.98	0.2039	79.44 ± 5.74
	PubMed	29.95	80.76	81.73	0.2486	79.81 ± 5.17
GT	Cora	21.80	37.81	38.08	0.9990	37.38 ± 1.69
	CiteSeer	27.56	37.78	37.88	0.9933	37.61 ± 3.04
	PubMed	39.23	71.17	73.45	0.9973	68.94 ± 0.94

effectiveness in tuning graph prompts to solve the respective downstream tasks. This highlights that our `DP-GPL` is effective in improving privacy-utility trade-offs.

Regarding our methods' privacy consumption, we observe that neither exhausts the given privacy budget of $\varepsilon = 2$. In particular, `DP-GPL` is not able to spend above $\varepsilon = 0.3627$ during the labeling. This small privacy consumption is due to the limited number of public samples used for the knowledge transfer: over the given 50 queries, the methods cannot spend more privacy. While it would be possible to increase the number of public queries, we find that this does not increase the downstream performance notably. Hence, by limiting the public data to 50 samples, the best privacy-utility trade-offs can be achieved.

6 Conclusions

In this work, we are the first to highlight the privacy risks that arise from graph prompt learning. By running a membership inference attack, we showed that private information from the private dataset used to tune the graph prompts can leak to external parties who query the prompted GNN. To mitigate the resulting risk for the downstream data, we set out to design a private graph prompt learning algorithm. Motivated by our finding that the naive application of the DP-SGD algorithm, the standard to implement DP guarantees in machine learning, fails to yield good privacy-utility trade-offs, we designed `DP-GPL`, which builds on the PATE algorithm and performs a noisy knowledge transfer from teachers to a student prompt. We thoroughly analyzed the resulting utility and privacy implications and highlighted that our `DP-GPL` is able to yield strong utility at high privacy guarantees. Thereby, our work contributes towards leveraging the computational and utility benefits from graph prompt learning but without additional privacy risks for the downstream data.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Al-Rabeah, M.H., Lakizadeh, A.: Prediction of drug-drug interaction events using graph neural networks based feature extraction. Scientific Reports **12**(1), 15590 (2022)
3. Boenisch, F., Mühl, C., Rinberg, R., Ihrig, J., Dziedzic, A.: Individualized pate: Differentially private machine learning with individual privacy guarantees. Proceedings on Privacy Enhancing Technologies (2023)
4. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1897–1914. IEEE (2022)
5. Chen, H., Yin, H., Sun, X., Chen, T., Gabrys, B., Musial, K.: Multi-level graph convolutional networks for cross-platform anchor link prediction. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1503–1511 (2020)
6. Chen, Y., Li, T., Liu, H., Yu, Y.: Hide and seek (has): A lightweight framework for prompt privacy protection. arXiv preprint arXiv:2309.03057 (2023)
7. Daigavane, A., Madan, G., Sinha, A., Thakurta, A.G., Aggarwal, G., Jain, P.: Node-level differentially private graph neural networks. arXiv preprint arXiv:2111.15521 (2021)
8. Duan, H., Dziedzic, A., Papernot, N., Boenisch, F.: Flocks of stochastic parrots: Differentially private prompt learning for large language models. Advances in Neural Information Processing Systems **36** (2023)
9. Duan, H., Dziedzic, A., Yaghini, M., Papernot, N., Boenisch, F.: On the privacy risk of in-context learning. In: The 61st Annual Meeting Of The Association For Computational Linguistics (2023)
10. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
11. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
12. Fang, T., Zhang, Y., Yang, Y., Wang, C., Chen, L.: Universal prompt tuning for graph neural networks. Advances in Neural Information Processing Systems **36** (2024)
13. Ge, Q., Zhao, Z., Liu, Y., Cheng, A., Li, X., Wang, S., Yin, D.: Enhancing graph neural networks with structure-based prompt. arXiv preprint arXiv:2310.17394 (2023)
14. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: 2009 Ninth IEEE International Conference on Data Mining. pp. 169–178. IEEE (2009)
15. Hong, J., Wang, J.T., Zhang, C., Zhangheng, L., Li, B., Wang, Z.: Dp-opt: Make large language model your privacy-preserving prompt engineer. In: The Twelfth International Conference on Learning Representations (2024)
16. Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., Tang, J.: Graphmae: Self-supervised masked graph autoencoders. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 594–604 (2022)
17. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems **33**, 22118–22133 (2020)

18. Huang, Q., Ren, H., Chen, P., Kržmanc, G., Zeng, D., Liang, P.S., Leskovec, J.: Prodigy: Enabling in-context learning over graphs. Advances in Neural Information Processing Systems **36** (2024)
19. Kasiviswanathan, S.P., Nissim, K., Raskhodnikova, S., Smith, A.: Analyzing graphs with node differential privacy. In: Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings. pp. 457–476. Springer (2013)
20. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2022)
21. Li, Y., Tsai, Y.L., Yu, C.M., Chen, P.Y., Ren, X.: Exploring the benefits of visual prompting in differential privacy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5158–5167 (2023)
22. Liu, Z., Yu, X., Fang, Y., Zhang, X.: Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In: Proceedings of the ACM Web Conference 2023. pp. 417–428 (2023)
23. Lyu, X., Han, Y., Wang, W., Qian, H., Tsang, I., Zhang, X.: Cross-context backdoor attacks against graph prompt learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2094–2105 (2024)
24. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
25. Mueller, T.T., Paetzold, J.C., Prabhakar, C., Usynin, D., Rueckert, D., Kaissis, G.: Differentially private graph neural networks for whole-graph classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(6), 7308–7318 (2022)
26. Mueller, T.T., Usynin, D., Paetzold, J.C., Braren, R., Rueckert, D., Kaissis, G.: Differentially private guarantees for analytics and machine learning on graphs: A survey of results. Journal of Privacy and Confidentiality **14**(1) (2024)
27. Olatunji, I.E., Funke, T., Khosla, M.: Releasing graph neural networks with differential privacy guarantees. Transactions on Machine Learning Research (2023)
28. Panda, A., Wu, T., Wang, J., Mittal, P.: Differentially private in-context learning. In: The 61st Annual Meeting Of The Association For Computational Linguistics (2023)
29. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K.: Semi-supervised knowledge transfer for deep learning from private training data. stat **1050**, 3 (2017)
30. Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Erlingsson, U.: Scalable private learning with pate. In: International Conference on Learning Representations (2018)
31. Qian, C., Tang, H., Yang, Z., Liang, H., Liu, Y.: Can large language models empower molecular property prediction? arXiv preprint arXiv:2307.07443 (2023)
32. Sajadmanesh, S., Gatica-Perez, D.: Progap: Progressive graph neural networks with differential privacy guarantees. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. pp. 596–605 (2024)
33. Sajadmanesh, S., Shamsabadi, A.S., Bellet, A., Gatica-Perez, D.: {GAP}: Differentially private graph neural networks with aggregation perturbation. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 3223–3240 (2023)
34. Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., Sun, Y.: Masked label prediction: Unified message passing model for semi-supervised classification. arXiv preprint arXiv:2009.03509 (2020)
35. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)

36. Sun, M., Zhou, K., He, X., Wang, Y., Wang, X.: Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1717–1727 (2022)
37. Sun, X., Zhang, J., Wu, X., Cheng, H., Xiong, Y., Li, J.: Graph prompt learning: A comprehensive survey and beyond. arxiv 2023. arXiv preprint arXiv:2311.16534 (2023)
38. Sun, X., Cheng, H., Li, J., Liu, B., Guan, J.: All in one: Multi-task prompting for graph neural networks. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2120–2131 (2023)
39. Sun, X., Cheng, H., Liu, B., Li, J., Chen, H., Xu, G., Yin, H.: Self-supervised hypergraph representation learning for sociological analysis. IEEE Transactions on Knowledge and Data Engineering **35**(11), 11860–11871 (2023)
40. Sun, X., Yin, H., Liu, B., Meng, Q., Cao, J., Zhou, A., Chen, H.: Structure learning via meta-hyperedge for dynamic rumor detection. IEEE transactions on knowledge and data engineering **35**(9), 9128–9139 (2022)
41. Sun, X., Zhang, J., Wu, X., Cheng, H., Xiong, Y., Li, J.: Graph prompt learning: A comprehensive survey and beyond. arXiv preprint arXiv:2311.16534 (2023)
42. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: International Conference on Machine Learning. pp. 21076–21089. PMLR (2022)
43. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
44. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018)
45. Wu, Y., Wen, R., Backes, M., Berrang, P., Humbert, M., Shen, Y., Zhang, Y.: Quantifying privacy risks of prompts in visual prompt learning (2023)
46. Xia, J., Wu, L., Chen, J., Hu, B., Li, S.Z.: Simgrace: A simple framework for graph contrastive learning without data augmentation. In: Proceedings of the ACM Web Conference 2022. pp. 1070–1079 (2022)
47. Xiang, Z., Wang, T., Wang, D.: Preserving node-level privacy in graph neural networks. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 4714–4732. IEEE (2024)
48. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management. pp. 15–24 (2016)
49. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. pp. 40–48. PMLR (2016)
50. Zi, C., Zhao, H., Sun, X., Lin, Y., Cheng, H., Li, J.: Prog: A graph prompt learning benchmark. arXiv preprint arXiv:2406.05346 (2024)

A Appendix

A.1 Experimental Setup: Datasets

Table 4: **Statistics of datasets.** $|\mathcal{V}|, |\mathcal{E}|, m, |\mathbb{C}|$ denote the number of nodes, num of edges, dimension of a node feature vector, and number of classes, respectively.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	m	$\lvert\mathbb{C}\rvert$
ogbn-arxiv	$169,343$	$1,166,243$	128	40
Cora	$2,708$	$10,556$	$1,433$	7
CiteSeer	$3,327$	$9,104$	$3,703$	6
PubMed	$19,717$	$88,648$	500	3

In this paper, we focus on graph prompt learning for node-level tasks. Also, we consider the scenario where a GNN model is pretrained on a large graph by the model provider, and users apply it to a specific downstream task (a smaller graph) through graph prompt learning [38]. To simulate this scenario, we use ogbn-arxiv, which is a large-scale graph dataset, as the pre-training dataset. For the downstream tasks, we use Cora, CiteSeer, and PubMed [49]. The statistics of datasets are presented in Table 4.

A.2 Experimental Setup: Hyperparameters

The default hyperparameters used in the GNN pre-training phase are presented in Table 5. And Table 6 shows the parameters for Confident-GNMax used in `DP-GPL`.

Table 5: **Default hyperparameter setting for GNN pre-training.**

Type	Hyperparameter	Setting
GAT	Architecture	3 layers
	Hidden unit size	128
GCN	Architecture	3 layers
	Hidden unit size	128
Graph Transformer	Architecture	3 layers
	Hidden unit size	128
Training	Learning rate	0.001
	Optimizer	Adam
	Epochs	300
	Batch size	128

Table 6: **Parameters for Confident-GNMax.** (T - threshold, σ_1, σ_2 - noise parameters)

GNN model	Downsteream dataset	T	σ_1	σ_2
GAT	Cora	170	5	100
GAT	CiteSeer	170	5	50
GAT	PubMed	170	1	20
GCN	Cora	150	1	20
GCN	CiteSeer	180	1	20
GCN	PubMed	170	1	20
GT	Cora	150	10	100
GT	CiteSeer	150	5	50
GT	PubMed	170	5	100

A.3 Pseudocode for our `DP-GPL`

We here provide the pseudocode for our `DP-GPL` algorithm in Algorithm 2. This algorithm includes the main five steps in our methods, *i.e.,* private data partition, teacher prompts training, prompting pre-trained GNN model, aggregation, and student prompt training. In this algorithm, we highlight the difference between our methods and the standard PATE in blue.

A.4 Additional Experiments

Performance of Graph Prompt Learning One advantage of graph prompt learning is that in the few-shot setting, the downstream performance of graph prompt learning is comparable to or even better than the "pre-train & fine-tune" paradigm. We implement preliminary experiments to compare the downstream performance of graph prompt learning and the fine-tuning paradigm in a 5-shot setting, as shown in Table 7. As we can see, in most cases, the testing accuracy of graph prompt methods is close to or higher than that of the fine-tuning paradigm, making it reasonable to explore the privacy risk of graph prompt learning in the few-shot setting.

MIA Results Figure 5 and Figure 6 show our MIA on CiteSeer and PubMed datasets, respectively, with 1-5 shots of private data used in training prompts. As we can observe, our MIA has higher attack success with few shots.

Results of DP-SGD on Graph Prompt Learning Table 8 shows the performance of DP-SGD on graph prompt learning with different privacy budgets and numbers of shots. It is evident that the DP-SGD algorithm significantly degrades the downstream task's performance even at high privacy budgets. Only when the number of shots increases to 100, the DP-SGD algorithm can achieve a high utility. However, in the few-shot setting (*i.e.,* less than 50 shots), the DP-SGD algorithm fails to have a great privacy-utility trade-off.

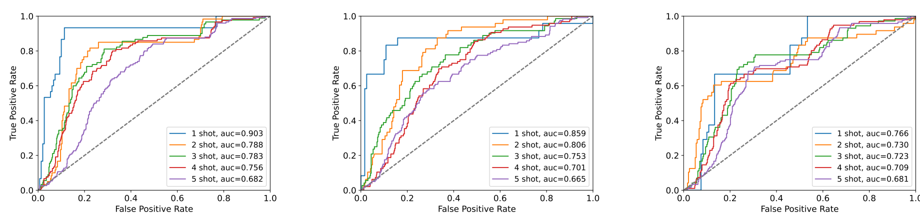
Algorithm 2 `DP-GPL`. In contrast to the standard PATE algorithm where the teacher models are trained on disjoint subsets of private data, our `DP-GPL` trains teacher prompts on disjoint subsets of the private graph data. We highlight these differences in blue.

Require: Private graph data $V_{private} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
Require: Number of teachers N, threshold T, noise parameters σ_1 and σ_2
Require: Pre-trained GNN model Φ, unlabeled public query data V_{public}

```
Step 1: Private data partition
/* DP-GPL */
Partition V_private into N IID disjoint groups {g_1, g_2, ..., g_N}
for each teacher i = 1 to N do
    Step 2: Teacher Prompts Training
    Train teacher prompt P_i on the group g_i
end for
Step 3: Prompting pre-trained GNN model
Actual public data D_public = ∅
for each query x_j ∈ V_public (e.g., a node) do
    Insert teacher prompt P_i into the query data point, i.e., P_i(x_j)
    Query the pre-trained GNN model and get a label y_i^j = Φ(P_i(x_j))
    Step 4: Aggregation
    /* DP-GPL */
    Get count for each class with uniform votes: c_m(x_j) = Σ_i^N (y_i^j = m)
    if max_m {c_m(x_j)} + N(0, σ_1^2) ≥ T then          ▷ m is the class label
        y^j = arg max_m {c_m(x_j) + N(0, σ_2^2)}
        D_public = D_public ∪ (x_j, y_j)
    end if
end for
Step 5: Student Prompt Training
Train student prompt P_s using the noisy labeled public data D_public
Differential Privacy Guarantee
Compute actual privacy loss (ε, δ) based on noise parameters σ_1, σ_2 and the number
  of queries |D_public|
return Student prompt P_s with differentially private guarantee
```

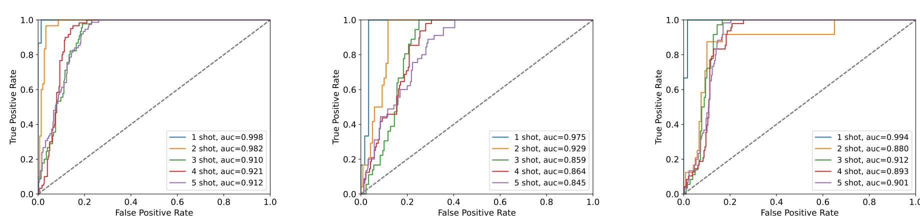
Table 7: **Performance of Pre-train & Fine-tune (PFT) and graph prompt learning (Cora, 5-shot).**

GNN architectures	Pre-train methods	PFT	All-in-one	GPF-plus	GPPT
GAT	DGI	46.03 ±0.79	48.70 ±1.45	53.48 ±1.99	56.53 ±1.51
	EdgePreGPPT	56.33 ±1.29	48.71 ±1.11	40.89 ±1.53	54.77 ±1.54
	GraphMAE	43.51 ±0.74	50.66 ±1.03	51.61 ±1.09	49.32 ±1.49
	SimGRACE	14.71 ±1.67	13.05 ±1.62	21.35 ±1.24	35.03 ±2.07
GCN	DGI	52.12 ±1.36	58.25 ±1.10	66.50 ±2.50	56.21 ±1.68
	EdgePreGPPT	43.77 ±1.16	68.94 ±1.09	76.30 ±0.98	60.28 ±1.86
	GraphMAE	39.55 ±1.24	62.90 ±0.91	75.84 ±1.10	51.63 ±1.25
	SimGRACE	18.15 ±0.52	18.19 ±1.64	19.97 ±0.65	33.72 ±1.98
GraphTransformer	DGI	53.33 ±1.09	45.12 ±2.05	29.54 ±2.24	56.21 ±1.51
	EdgePreGPPT	60.02 ±1.07	53.45 ±1.06	35.74 ±0.59	56.95 ±1.04
	GraphMAE	52.95 ±1.44	41.84 ±0.97	36.58 ±0.67	48.54 ±1.17
	SimGRACE	39.79 ±0.25	15.03 ±1.12	15.60 ±0.88	41.14 ±0.57



(a) GAT (b) GCN (c) Graph Transformer

Fig. 5: **AUC-ROC curve of our MIA on CiteSeer dataset with different number of shots, *i.e.,* 1-5 shots.**



(a) GAT (b) GCN (c) Graph Transformer

Fig. 6: **AUC-ROC curve of our MIA attack on PubMed dataset with different number of shots, *i.e.,* 1-5 shots.**

Additional `DP-GPL` Results We also present the performance of our `DP-GPL` on other setups, see Table 9 to Table 11. In consistent with the observations in Section 5.2, our `DP-GPL` can achieve high utility under strong privacy guarantees.

Table 8: **Performance of DP-SGD on graph prompt learning on Cora dataset (DGI, GPF-plus, GAT).**

# Shots	$\varepsilon=\infty$	$\varepsilon=1$	$\varepsilon=8$	$\varepsilon=16$	$\varepsilon=32$	$\varepsilon=64$
5	48.70 ±1.45	15.10 ±1.09	15.46 ±1.13	16.58 ±0.17	17.04 ±1.01	18.47 ±0.91
10	65.70 ±5.15	17.04 ±0.43	16.75 ±3.29	17.33 ±2.91	18.09 ±0.21	18.67 ±0.96
50	75.20 ±2.09	19.58 ±0.17	19.91 ±3.15	22.55 ±1.63	22.44 ±2.04	22.04 ±1.20
100	78.42 ±0.98	68.15 ±0.94	77.27 ±0.33	77.94 ±1.85	78.16 ±1.94	78.40 ±1.53

Table 9: **Performance comparison between our `DP-GPL` and three baselines on three downstream datasets. (GraphMAE, All-in-one, $\delta=1.5\times10^{-4}$).** LB – Lower Bound, UB – Upper Bound.

		LB	Ens. Acc.	UB	our `DP-GPL`	
	Private	$\varepsilon=0$	$\varepsilon=\infty$	$\varepsilon=\infty$	ε	Test Acc
GAT	Cora	39.65	49.40	52.94	0.5728	41.02 ±1.38
	CiteSeer	38.50	39.09	40.87	0.2412	29.27 ±2.10
	PubMed	30.86	64.64	67.85	0.2232	58.81 ±0.59
GCN	Cora	30.76	62.97	65.37	0.0782	59.50 ±0.63
	CiteSeer	31.85	67.89	71.85	0.0588	61.68 ±0.41
	PubMed	32.87	70.22	71.46	0.4989	64.59 ±0.11
GT	Cora	35.68	47.35	48.65	0.4197	37.47 ±1.05
	CiteSeer	34.67	52.58	56.48	0.0390	46.97 ±2.18
	PubMed	22.38	34.34	35.47	0.3359	32.82 ±1.42

Influence of the number of queries We analyze the impact of the number of public queries on the performance of our `DP-GPL` in Figure 7, taking Cora, DGI, All-in-one, and GAT as an example. As we can see, the performance of our `DP-GPL` increases as the number of public queries increases from 10 to 50. With more than 50 public queries, the performance of our `DP-GPL` tends to be stable, indicating that our methods can achieve the best privacy-utility trade-offs with 50 public queries.

MIA results against `DP-GPL` We also evaluate the effectiveness of our `DP-GPL` against MIA, as shown in Figure 8. The member data is the private data used in training all teacher prompts, and the non-members are randomly selected samples from the testing dataset. As we can see, all curves are very close to the dash line (random guess), which shows that our `DP-GPL` is effective against MIA, for all downstream tasks and GNN architectures.

Table 10: **Performance comparison between our DP-GPL and three baselines on three downstream datasets. (GraphMAE, GPF-plus, $\delta = 1.5 \times 10^{-4}$).** LB – Lower Bound, UB – Upper Bound.

		LB	Ens. Acc.	UB	our DP-GPL	
	Private	$\varepsilon = 0$	$\varepsilon = \infty$	$\varepsilon = \infty$	ε	Test Acc
GAT	Cora	39.65	51.69	54.38	0.6778	45.44 ± 7.09
	CiteSeer	38.50	58.02	61.94	0.2194	54.50 ± 3.41
	PubMed	30.86	76.21	78.56	0.4846	66.21 ± 3.05
GCN	Cora	30.76	74.16	76.85	0.6135	66.88 ± 1.91
	CiteSeer	31.85	78.13	80.87	0.6262	69.45 ± 2.58
	PubMed	32.87	77.84	80.85	0.0595	68.67 ± 5.32
GT	Cora	35.68	39.10	42.49	0.0273	30.78 ± 3.33
	CiteSeer	34.67	41.61	43.75	0.1189	34.72 ± 0.54
	PubMed	22.38	28.29	31.39	0.6147	21.86 ± 1.69

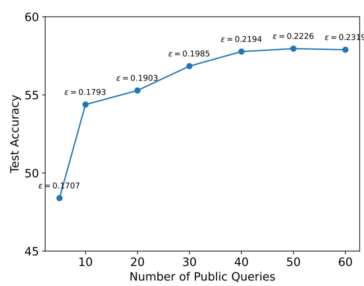
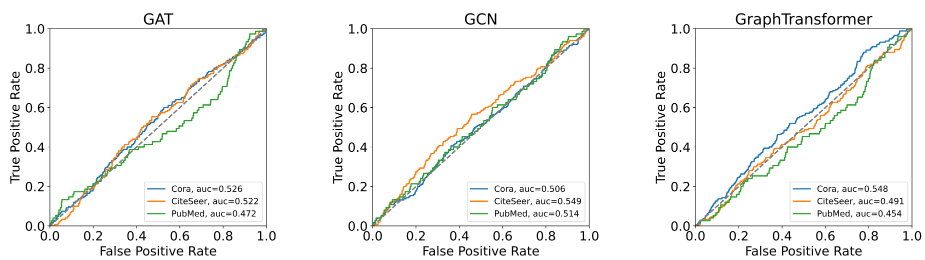


Fig. 7: **Influence of the number of public queries on the performance of our DP-GPL (Cora, DGI, All-in-one, GAT).**

Table 11: **Performance comparison between our DP-GPL and three baselines on three downstream datasets. (GraphMAE, GPPT, $\delta = 1.5\times10^{-4}$).** LB – Lower Bound, UB – Upper Bound.

		LB	Ens. Acc.	UB	our DP-GPL	
	Private	$\varepsilon = 0$	$\varepsilon = \infty$	$\varepsilon = \infty$	ε	Test Acc
GAT	Cora	39.65	49.99	51.57	0.5979	47.08 ± 2.13
	CiteSeer	38.50	45.44	46.48	0.6392	43.02 ± 0.41
	PubMed	30.86	55.48	56.64	0.5325	53.92 ± 0.05
GCN	Cora	30.76	54.57	54.85	0.3617	51.61 ± 3.98
	CiteSeer	31.85	44.12	45.78	0.1175	41.53 ± 1.17
	PubMed	32.87	59.25	60.57	0.2091	56.35 ± 1.64
GT	Cora	35.68	52.63	54.09	0.1988	50.24 ± 4.11
	CiteSeer	34.67	65.16	65.78	0.2290	63.49 ± 5.21
	PubMed	22.38	46.41	47.97	0.3221	44.03 ± 3.00



(a) GAT (b) GCN (c) Graph Transformer

Fig. 8: **AUC-ROC curve of our MIA against** `DP-GPL` **(Cora, 5 shots).** Generally, all curves are very close to the dash line (random guess), which shows that `DP-GPL` is effective against MIA.