Poincaré Differential Privacy for Hierarchy-Aware Graph Embedding

Yuecen Wei[1,2,3], **Haonan Yuan**[1], **Xingcheng Fu**[3*], **Qingyun Sun**[1], **Hao Peng**[1], **Xianxian Li**[3], **Chunming Hu**[1,2*]

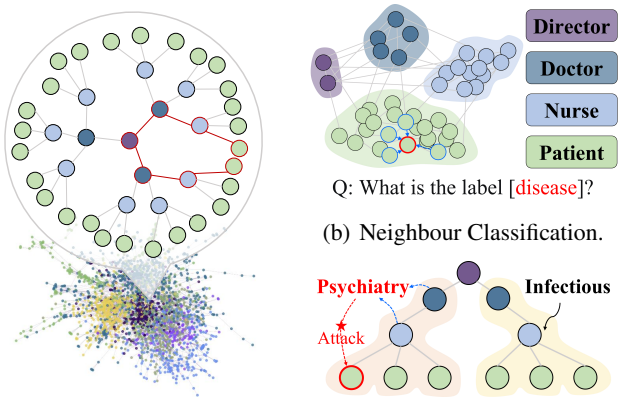
[1]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
[2]School of Software, Beihang University, Beijing, China
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
{weiyc,yuanhn,sunqy,penghao,hucm}@buaa.edu.cn, {fuxc,lixx}@gxnu.edu.cn

Abstract

Hierarchy is an important and commonly observed topological property in real-world graphs that indicate the relationships between supervisors and subordinates or the organizational behavior of human groups. As hierarchy is introduced as a new inductive bias into the Graph Neural Networks (GNNs) in various tasks, it implies latent topological relations for attackers to improve their inference attack performance, leading to serious privacy leakage issues. In addition, existing privacy-preserving frameworks suffer from reduced protection ability in hierarchical propagation due to the deficiency of adaptive upper-bound estimation of the hierarchical perturbation boundary. It is of great urgency to effectively leverage the hierarchical property of data while satisfying privacy guarantees. To solve the problem, we propose the **Poin**caré **D**ifferential **P**rivacy framework, named **PoinDP**, to protect the hierarchy-aware graph embedding based on hyperbolic geometry. Specifically, PoinDP first learns the hierarchy weights for each entity based on the Poincaré model in hyperbolic space. Then, the Personalized Hierarchy-aware Sensitivity is designed to measure the sensitivity of the hierarchical structure and adaptively allocate the privacy protection strength. Besides, the Hyperbolic Gaussian Mechanism (HGM) is proposed to extend the Gaussian mechanism in Euclidean space to hyperbolic space to realize random perturbations that satisfy differential privacy under the hyperbolic space metric. Extensive experiment results on five real-world datasets demonstrate the proposed PoinDP's advantages of effective privacy protection while maintaining good performance on the node classification task.

Introduction

The inherent topological properties of graphs have been widely leveraged in graph representation learning as inductive biases (Sun et al. 2022c; Li et al. 2023; Zhang et al. 2024; Yuan et al. 2024). Real-world graph data typically exhibit intricate topological structures with diverse properties (Sun et al. 2021b), and the **hierarchy** frequently assumes a pivotal role, which naturally mirrors human behavior within hierarchical organizations. This property assists in the learning of graph representation by capturing implicit data organization patterns (Papadopoulos et al. 2012).



(a) Latent Tree Structure. (b) Neighbour Classification. (c) Hierarchical Classification.

Figure 1: Privacy leakage on the hierarchical structure.

However, dealing directly with hierarchy in the topological space proves to be challenging in the Euclidean embedding space. In contrast to the Euclidean space, the hyperbolic geometric space can be conceptualized as a continuous tree-like structure, naturally capable of representing the topological hierarchy (Sun et al. 2022a, 2023b). The constant negative curvature of the hyperbolic geometric space imparts it with a more potent ability for hierarchical representation compared to the flat Euclidean space (Krioukov et al. 2010; Sun et al. 2023a). Recent works in hyperbolic representation learning (Ganea, Bécigneul, and Hofmann 2018; Tifrea, Bécigneul, and Ganea 2019) have achieved noteworthy success by harnessing the hierarchy, where the hierarchy is regarded as the prompt for balancing the aggregation weights. These methods manifest how the hierarchy can be utilized to enhance the effectiveness of graph representation learning.

However, while representation learning in the hyperbolic geometric space offers benefits, it comes with the potential drawback of increased susceptibility to the leakage of sensitive user information. Hierarchical structures are prevalent in graph data, as depicted in Figure 1(a). Euclidean and hyperbolic spaces exhibit distinct perceptual capabilities for hierarchical structures. For instance, in Euclidean space, clustering based on node distances can reflect whether an individual is a patient, while node distances in hyperbolic space

*Corresponding authors

can indicate the type of ailment. In addition, traditional Euclidean Graph Neural Networks (GNNs) primarily focus on neighborhood aggregation and struggle to capture latent hierarchical tree-like structures, as illustrated in Figure 1(b). Consequently, the feature perturbation conducted by conventional GNN privacy frameworks only disturbs connections among nodes of the same category, to thwart attacker inferences. However, **although hyperbolic GNNs provide a direct approach to learning hierarchical structural features, they concurrently elevate the risk of privacy leakage**, as depicted in Figure 1(c). Attackers, without accessing sensitive information, can deduce patients' medical conditions. For example, within the hierarchical structure, an attacker can infer that patients affiliated with psychiatric departments are likely to have mental illnesses, and patients in infectious disease departments have contagious diseases.

To address privacy concerns, notable privacy-preserving techniques have been proposed. Differential privacy (DP) (Dwork 2006; Dwork et al. 2006) stands out as one of the most prominent methods due to its robust mathematical foundation. However, existing DP methods tailored for graph representation learning have predominantly centered on safeguarding node features and neighborhood structures (Ren et al. 2022; Yang et al. 2021; Wei et al. 2022), with a limited focus on preserving implicit topological properties such as hierarchy. This underscores the need for novel strategies that holistically address both the intricate topological features and privacy considerations within graph representation learning.

To utilize the geometric prior of the hyperbolic space to capture the hierarchy properties and guarantee that the sensitive information in the hierarchy, the major problems are as follows: (1) Traditional privacy-preserving methods usually consider the privacy between neighbors or relations to generate perturbation noise, which is weak to capture the hierarchical structure of the graph. (2) Existing privacy-preserving techniques measure the privacy of nodes in Euclidean space, which doesn't work in hyperbolic space due to the Gaussian mechanism based on the standard normal distribution just defined in Euclidean space.

Present work. To address the above problems, we propose a novel **Poin**caré **D**ifferential **P**rivacy framework for protecting hierarchy-aware graph embedding based on hyperbolic geometry, named **PoinDP**[1]. First, the Personalized Hierarchy-aware Sensitivity (PHS) is designed to utilize the Poincaré model to capture the inter- and intra-hierarchy node information. PHS can allocate the privacy budget between *radius* (inter-hierarchy) and *angle* (intra-hierarchy) and learn high-quality graph representations effectively while satisfying the differential privacy guarantee. Then, a novel Hyperbolic Gaussian Mechanism (HGM) extends the Gaussian mechanism in Euclidean space to hyperbolic space to realize random perturbations that satisfy differential privacy under the hyperbolic space metric for the first time. Extensive experimental results conducted on five datasets empirically demonstrate that PoinDP has consistent advantages. We summarize our contributions as follows:

- We propose a novel Poincaré differential privacy for hierarchy-aware graph embedding framework named PoinDP. To the best of our knowledge, this is the first work that presents the privacy leakage problem due to the hierarchical structure and gives a definition of the privacy problem in terms of hyperbolic geometry.
- In PoinDP, the Personalized Hierarchy-aware Sensitivity can measure the sensitivity of the hierarchical structure and adaptively allocate the privacy protection strength. Besides, we extend the Gaussian mechanism in Euclidean space to hyperbolic space to realize random perturbations that satisfy differential privacy under the hyperbolic space metric for the first time, which can be used in other hyperbolic privacy works to promote community development.
- Experiments demonstrate that PoinDP can effectively resist attackers with hierarchical information enhancement, and learn high-quality graph representations while satisfying privacy guarantees.

Related Work

Graph Neural Networks

In the field of graph representation learning, Graph Neural Networks (GNNs) have achieved remarkable success in learning embeddings from graph-structured data due to their powerful graph representation capabilities, while are widely extended for downstream tasks in complex scenarios (Kipf and Welling 2017; Velickovic et al. 2018; Hamilton, Ying, and Leskovec 2017). However, traditional GNNs operating in Euclidean space often fall short of effectively utilizing the topology properties of graphs, leading to suboptimal semantic understanding, particularly overlooking the hierarchical relationships within the data, which is of vital importance in real-world scenarios.

Recently, certain categories of data (e.g., hierarchical, scale-free, or spherical data) have demonstrated superior representation capabilities when modeled through non-Euclidean geometries. This has led to a burgeoning body of work on deep learning (Tifrea, Bécigneul, and Ganea 2019; Sala et al. 2018; Ganea, Bécigneul, and Hofmann 2018). Notably, hyperbolic geometric spaces have garnered significant attention and adoption within the domain of graph representation learning (Liu, Nickel, and Kiela 2019; Chami et al. 2019; Bachmann, Bécigneul, and Ganea 2020; Sun et al. 2021a; Fu et al. 2023; Sun et al. 2022d; Wu et al. 2022), attributed to their inherent capacity and prowess in preserving hierarchical structures (Sun et al. 2022b).

However, with the evolution of increasingly intricate models aimed at extracting potential correlations among nodes, the complex structure inadvertently amplifies the attackers' capacity for inference, enabling lateral enhancement of their inferential ability. Unfortunately, the majority of GNN-based methodologies have been demonstrated to possess vulnerabilities susceptible to inference attacks (Olatunji, Nejdl, and Khosla 2021; Zhang et al. 2022).

Differentially Private GNNs

Differential privacy (DP) (Dwork 2006) is a privacy protection method and introduces random noise perturbation

[1]Code is available at https://github.com/WYLucency/PoinDP.

mechanisms to the original data, ensuring that attackers cannot infer the original data from the outputs of models. For graph privacy protection, we divide the existing DP method into two levels: node-level and edge-level.

For node-level DP, the works focus on perturbing node features or node labels to execute privacy protection. AsgLDP (Wei et al. 2020) proposed randomized attribute lists (RAL) to perturb each bit of node feature by the randomized response, and LPGNN (Sajadmanesh and Gatica-Perez 2021) used a multi-bit mechanism to sample perturbed features while using the randomized response to mask node labels. GAP (Sajadmanesh et al. 2023) perturbed the out of each aggregation using Gaussian noise while saving them. HeteDP (Wei et al. 2022) utilized meta-path to adapt data heterogeneity while personalized node perturbation by multi-attention mechanism.

For edge-level DP, the target of noise addition is the topology of the graph, e.g. the degree and the adjacency matrix that represent the information about the interactions between nodes. LDPGEN (Qin et al. 2017) computed each subgroup degree vector on the client, then uploaded it to the server and exerted Laplace noise to the degree vectors. The graph structure generation uses the BTER model. Solitude (Lin, Li, and Wang 2022) used the randomized response to flip graph adjacency matrix and a regularization term to optimize noise. LF-GDPR (Ye et al. 2022) perturbed node degrees and adjacency matrix in the client and the server will receive a double-degree message to aggregate and calibrate.

However, most DP schemes are deficient in adaptability to complex structures and hardly explore potential properties to adjust perturb design.

Preliminary

Differential privacy (Dwork 2006) is considered to be one of the quantifiable and practical privacy-preserving data processing techniques. It protects privacy by adding noise to the query results, and an attacker cannot infer any information from these query results even if he or she knows all the records except this particular individual information.

Definition 1 ((ϵ, δ)-Differential Privacy)**.** *Given two adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$ differ by at most one record, and they are protected via a random algorithm $\mathcal{M}$, which satisfies (ϵ, δ)-differential privacy (DP) (Dwork et al. 2006). For any possible subset of output $\mathcal{O} \subseteq \mathrm{Range}(\mathcal{M})$, we have*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq \mathrm{e}^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta, \quad (1)$$

where ϵ is the privacy budget, δ is a probability to break ϵ-DP and $\mathrm{Range}(\mathcal{M})$ denotes the value range of $\mathcal{M}$ output.

Definition 2 (Sensitivity)**.** *Given any query $\mathcal{S}$ on D, the sensitivity (Dwork et al. 2006) for any neighboring datasets D and D' are defined as*

$$\Delta_2 \mathcal{S} = \max_{D, D'} \|\mathcal{S}(D) - \mathcal{S}(D')\|_2 . \quad (2)$$

Definition 3 (Gaussian Mechanism)**.** *Let $\mathcal{S} : D \to \mathbf{O}^{\mathcal{K}}$ be an arbitrary $\mathcal{K}$-dimensional function and define its L_2 sensitivity to be $\Delta_2 \mathcal{S}$. The Gaussian Mechanism (Dwork and Roth 2014) with parameter σ adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the n components of the output. Given $\epsilon \in (0, 1)$, the Gaussian Mechanism is (ϵ, δ)-DP with*

$$\sigma \geq \sqrt{2 \ln (1.25/\delta)} \Delta_2 \mathcal{S} / \epsilon. \quad (3)$$

For a graph $\mathcal{G}$, the overall form of the perturbed noise is defined as

$$\mathcal{M}(\mathcal{G}) \triangleq \mathcal{S}(\mathcal{G}) + \mathcal{N}\left(0, (\triangle_2 \mathcal{S})^2 \sigma^2\right), \quad (4)$$

where $\Delta_2 \mathcal{S}$ controls the amount of noise in the generated Gaussian distribution from which we will sample noise into the target.

Our goal is to keep (ϵ, δ)-DP effective in high-dimensional projection spaces and message passing while maintaining classification performance. Compared to the traditional Euclidean space, hyperbolic space has a stronger hierarchical structure. The Poincaré ball model (Nickel and Kiela 2017) is a commonly used isometric model in hyperbolic space, and we exploit it to capture the latent hierarchical structure of the graph.

Definition 4 (Poincaré Ball Model)**.** *Given a constant negative curvature c, Poincaré Ball $\mathcal{B}^n$ is a Riemannian manifold $(\mathcal{B}_c^n, g_x^{\mathcal{B}})$, where $\mathcal{B}_c^n$ is an n-dimensional ball of radius $1/\sqrt{c}$ and $g_x^{\mathcal{B}}$ is metric tensor. The Poincaré distance between the node pair $(\mathbf{x}, \mathbf{y})$ is defined as*

$$d_{\mathcal{B}_c^n}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \, \|-\mathbf{x} \oplus_c \mathbf{y}\|), \quad (5)$$

where $\oplus_c$ is Möbius addition and $\|\cdot\|$ is L_2 norm.

Definition 5 (Poincaré Norm)**.** *The Poincaré Norm is defined as the distance of any point $\mathbf{x} \in \mathcal{B}_c^n$ from the origin of Poincaré ball:*

$$\mathrm{Norm}_{\mathcal{B}_c^n}(\mathbf{x}) = \|\mathbf{x}\|_{\mathcal{B}_c^n} = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \, \|\mathbf{x}\|). \quad (6)$$

Our Approach

In this section, we introduce the overall learning framework of PoinDP, a unified hierarchy-aware graph neural network for privacy guarantees with differential privacy, and find out how to achieve privacy protection in the hierarchy structure. The framework is shown in Figure 2, and the overall process of PoinDP is shown in Algorithm 1.

Personalized Hierarchy-aware Sensitivity

To the best of our knowledge, as described in many existing privacy-preserving models, the sensitivity of differential privacy is usually measured by the L_2 norm (Euclidean distance), which makes it difficult to measure the non-Euclidean structure accurately. Therefore, the sensitivity of the traditional method measured under Euclidean space is not accurate when used in the hierarchy. Moreover, to solve the personalized privacy requirements of hierarchical structures, we aim to be able to explicitly design for inter- and intra-hierarchy sensitivity. Inspired by the hyperbolic geometric prior, we design a novel *Personalized Hierarchy-aware Sensitivity* based on the Poincaré embedding (Nickel

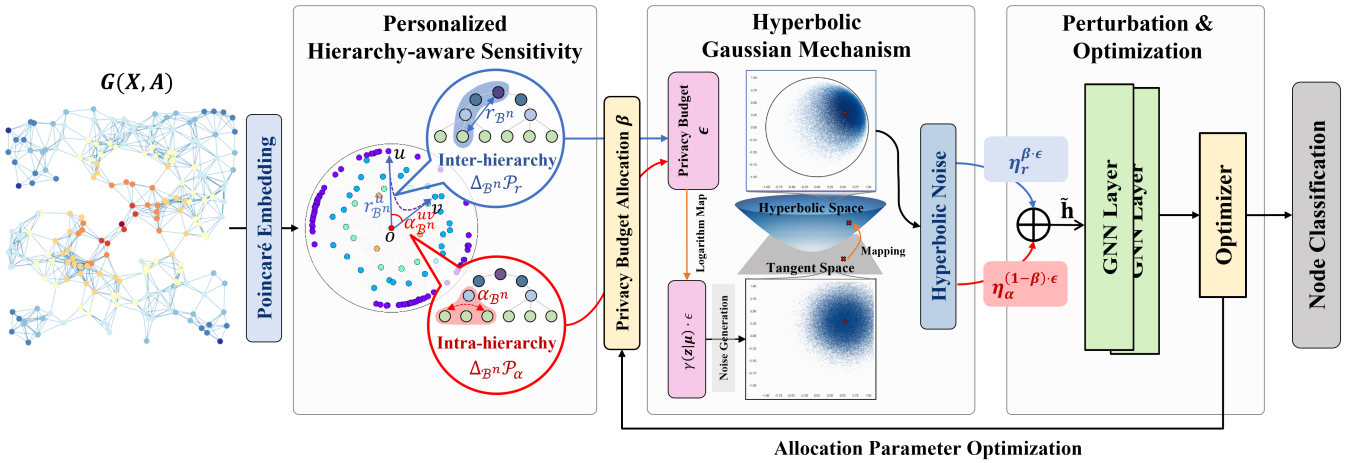


Figure 2: Overview of PoinDP. Given $\mathcal{G}$ as input, PoinDP consists of the following three steps: (1) PHS Computing: We first obtain the Poincaré embedding using the adjacency matrix A and compute the PHS. (2) Noise generation: The sensitivity is utilized to perform the HGM in order to obtain the hyperbolic noise satisfying (ϵ, δ)-DP. (3) Perturbation & Optimization: The noise is injected into GNNs, and the privacy budget allocation is optimized according to the downstream task feedback.

and Kiela 2017) for generating random perturbation noise with adaptive inter- and intra-hierarchy correlations.

Hierarchy-aware node representation. First, we need to explicitly represent the graph hierarchy based on the Poincaré embedding. We utilize the Poincaré embedding to learn a hierarchy-aware node representation, which is a shallow model, and minimize a hyperbolic distance-based loss function. Then We learn node embeddings $\mathbf{e}^V = \{\mathbf{e}_i\}_{i=1}^{\|\mathcal{V}\|}$ $(\mathbf{e}_i \in \mathcal{B}^n, V \in \mathcal{V})$[2] which represents the hierarchy of nodes in the Poincaré ball model based on Eq. (5). The embeddings can be optimized as

$$\mathbf{\Theta}' \leftarrow \underset{\mathbf{\Theta}}{\operatorname{argmin}} \mathcal{L}(\mathbf{\Theta}), \quad \mathbf{s.t.}\ \forall \theta_i \in \mathbf{\Theta} : \|\theta_i\| < 1/c, \tag{7}$$

where $\mathcal{L}(\mathbf{\Theta})$ is a softmax loss function that approximates the dependency between nodes, $\mathbf{\Theta}$ is the parameters of Poincaré ball model.

We can obtain the radius and angle of the node on the Poincaré disk based on the Poincaré embedding $\mathbf{e}^V$. A smaller $\mathrm{Norm}_{\mathcal{B}^n}(\mathbf{e}_u)$ indicates that u is located at the top of the hierarchy. The node with a top-level hierarchy is approximately near the center of the disk and plays a more important role in the graph. Then we can give the inter-hierarchy sensitivity by using the radius r of nodes on Poincaré disk.

Definition 6 (Inter-hierarchy Sensitivity)**.** *Given V and V' are the neighboring subsets of graph nodes, and V and V' only differ by one node. The inter-hierarchy sensitivity can be defined as:*

$$\Delta_{\mathcal{B}^n}\mathcal{P}_r = \max_{V,V'} \left| \mathrm{Norm}_{\mathcal{B}^n}(\mathbf{e}^V) - \mathrm{Norm}_{\mathcal{B}^n}(\mathbf{e}^{V'}) \right|. \tag{8}$$

On the other hand, since the angle sector on the hyperbolic disk indicates the node similarity or the community, we use it to measure the correlations of nodes within a hierarchy level. As the Poincaré ball is conformal to Euclidean space (Ganea, Bécigneul, and Hofmann 2018), the angle between two vector u, v at the radius r is given by

$$\alpha(u,v)|_{\mathbf{e}^V} = \frac{g_x^{\mathcal{B}^n}(u,v)}{\sqrt{g_x^{\mathcal{B}^p}(u,u)}\sqrt{g_x^{\mathcal{B}^n}(v,v)}} = \frac{\langle \mathbf{e}_u, \mathbf{e}_v \rangle}{\|\mathbf{e}_u\|\,\|\mathbf{e}_v\|}. \tag{9}$$

Similarly, we measure the correlation within the intra-hierarchy based on the angle α between any two nodes on the Poincaré disk.

Definition 7 (Intra-hierarchy Sensitivity)**.** *Given V and V' are the neighboring subsets of graph nodes, and V and V' only differ by one node. The intra-hierarchy sensitivity can be defined as:*

$$\Delta_{\mathcal{B}^n}\mathcal{P}_\alpha = \max_{V,V'} \|\alpha(V,V')|_{\mathbf{e}^{(V\cup V')}}\|_{\mathcal{B}^n}. \tag{10}$$

Then we utilize the inter-hierarchy $\Delta_{\mathcal{B}^n}\mathcal{P}_r$ and intra-hierarchy $\Delta_{\mathcal{B}^n}\mathcal{P}_\alpha$ sensitivities separately to focus on the importance of nodes at different radius and angles and generate perturbation noises that satisfy personalization.

Hyperbolic Gaussian Mechanism

The existing works widely used differential privacy strategies based on Laplace noise or Gaussian noise to achieve protection. However, due to the difference in metric scales, their noise computation can only be performed in flat Euclidean space, which is difficult to adapt to curved hyperbolic space. To address the privacy issues proposed by the hierarchy of graphs, we design a *Hyperbolic Gaussian Mechanism* that will extend the Gaussian mechanism in Euclidean space to hyperbolic space based on the *Wrapped Gaussian Distribution* (Nagano et al. 2019) to realize stochastic perturbations that satisfy differential privacy in the metric of hyperbolic space. The hyperbolic Gaussian

[2]We use the Poincaré ball model with standard constant negative curvature $\|c\| = 1$, the curvature parameter c will be omitted in our method.

distribution with $c = 1$ is defined as

$$\mathcal{N}_{\mathcal{B}^n}(\mathbf{z}|\mu, \sigma_\epsilon^2\mathbf{I}) = \mathcal{N}(\lambda_\mu \log_\mu(\mathbf{z})|\mathbf{0}, \sigma_\epsilon^2\mathbf{I}) \cdot \gamma((\mathbf{z}|\mu)),$$
$$\text{with } \gamma(\mathbf{z}|\mu) = \left(\frac{d_{\mathcal{B}^n}(\mu, \mathbf{z})}{\sinh d_{\mathcal{B}^n(\mu,\mathbf{z})}}\right)^{n-1}, \tag{11}$$

where $\mu \in \mathcal{B}^n$ is mean parameter, $\sigma_\epsilon \in \mathbf{R}^n$ is standard deviation, $\log_\mu(\cdot)$ is the logarithm map function,and γ represents the spatial mapping and normalization.

Hyperbolic Gaussian Mechanism. Let $f : \mathcal{B}^{|\mathcal{X}|} \to \mathbf{R}^n$ be an arbitrary n-dimensional function, and define its hyperbolic sensitivity to be $\Delta_{\mathcal{B}^n} f = \max_{\text{adjacent}(D,D')} \|f(D) - f(D')\|_{\mathcal{B}^n}$. The *Hyperbolic Gaussian Mechanism* with parameters σ adds noise scaled to $\mathcal{N}_{\mathcal{B}^n}(\cdot|0, \sigma^2\mathbf{I})$ to each of the n components of the output.

Theorem 1. *Let $\epsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln(1.25\gamma(\cdot|\mu)/\delta)$, the Hyperbolic Gaussian Mechanism with parameter $\sigma \geq c \log_\mu(\Delta_{\mathcal{B}^n} f)\gamma(\cdot|\mu)/\epsilon$ is (ϵ, δ)-differentially private on hyperbolic space.*

To satisfy the (ϵ, δ)-differentially private in hyperbolic space, the hyperbolic sensitivity and hyperbolic Gaussian noise sampling need to be mapped to the tangent space by logarithm map function $\log_\mu(\cdot)$, and the privacy budget ϵ and parameter δ also need to be isometric mapped in the tangent space of μ. Please refer to Appendix A.1 for the detailed proof.

According to the above, we can obtain two kinds of perturbation noise based on the inter-hierarchy $\Delta_{\mathcal{B}^n}\mathcal{P}_r$ and intra-hierarchy $\Delta_{\mathcal{B}^n}\mathcal{P}_\alpha$ sensitivities. The Hyperbolic Gaussian Noise can be generated by

$$\begin{aligned}\eta_r^{\epsilon_r} &\sim \mathcal{N}_{\mathcal{B}^n}(\mathbf{z}|\mu, c^2 \log_\mu(\Delta_{\mathcal{B}^n}\mathcal{P}_r)^2\gamma(\mathbf{z}|\mu)^2/\epsilon_r^2\mathbf{I}),\\ \eta_\alpha^{\epsilon_\alpha} &\sim \mathcal{N}_{\mathcal{B}^n}(\mathbf{z}|\mu, c^2 \log_\mu(\Delta_{\mathcal{B}^n}\mathcal{P}_\alpha)^2\gamma(\mathbf{z}|\mu)^2/\epsilon_\alpha^2\mathbf{I}).\end{aligned} \tag{12}$$

Perturbation and Optimization

To better utilize hierarchical information to provide hierarchy-aware privacy perturbations, we utilize GNNs to capture the domain representation of nodes. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$. For the semi-supervised node classification task, given the labeled node set $\mathcal{V}_L$ and their labels $\mathcal{Y}_L$, where each node v_i is mapped to a label y_i, our goal aims to train a node classifier f_θ to predict the labels $\mathcal{Y}_U$ of remaining unlabeled nodes $\mathcal{V}_U = \mathcal{V} \setminus \mathcal{V}_L$. Therefore, following the aggregation and update mechanism of message passing in GNNs, we define the embedding learning of nodes u in $(l+1)$-th layer as

$$\mathbf{h}_u^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{V}(u)} c_v \mathbf{W}^{(l)} \mathbf{h}_v^{(l)}\right), \tag{13}$$

where c_v is a node-wise normalization constant and $\mathcal{V}(u)$ is the neighbor set. During the continuous iteration in the training stage, the features of node u will be updated with a hyperbolic Gaussian mechanism as

$$\hat{\mathbf{h}} = \mathbf{h} + \eta_r^{\beta\cdot\epsilon} + \eta_\alpha^{(1-\beta)\cdot\epsilon}, \tag{14}$$

where β is the normalized attention weight to learn the inter- and intra-hierarchy importance in nodes and rationally allocate the privacy budget, i.e. $\epsilon_r + \epsilon_\alpha = \epsilon$.

Algorithm 1: Overall training process of PoinDP

Input: Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node labels $\mathcal{Y}$; Number of training epochs E.
Output: Predicted label $\hat{\mathcal{Y}}$.

Parameter Θ initialization;
Learning and optimizing node Poincaré embedding $\mathbf{e}^V \leftarrow$ Eq. (7) and (9);
for $e = 1, 2, \cdots, E$ **do**
  `// Personalized Hierarchy-aware Sensitivity`
  Calculate hierarchy-aware sensitivity $\Delta_{\mathcal{B}^n}\mathcal{P}_r$ and $\Delta_{\mathcal{B}^n}\mathcal{P}_\alpha \leftarrow$ Eq. (8) and (10);
  `// Hyperbolic Gaussian Mechanism`
  Calculate the hyperbolic Gaussian distribution $\mathcal{N}_{\mathcal{B}^n}(\mathbf{z}|\mu, \sigma_\epsilon^2\mathbf{I}) \leftarrow$ Eq. (11);
  Learning node embeddings $\mathbf{h}_u \leftarrow$ Eq. (13);
  Perturbing node embeddings $\widehat{\mathbf{h}}$ by hyperbolic Gaussian noise $\leftarrow$ Eq. (14);
  Predict node labels $\hat{\mathcal{Y}}$ and calculate the classification loss $\mathcal{L} \leftarrow$ Eq. (15);
  Update model parameters by minimizing $\mathcal{L}$.
end

Therefore, we complete the noise generation and addition by hierarchy-aware mechanism. The objective for PoinDP is the average loss of predicting labels of unlabeled nodes, formulated as

$$\mathcal{L} = \frac{1}{\|\mathcal{V}_U\|} \sum_{v \in \mathcal{V}_U} \mathcal{L}_\mathcal{G}(\hat{\mathbf{h}}_{u,v}, y_v), \tag{15}$$

where $\mathcal{L}_\mathcal{G}$ stands for the loss of semi-supervised node classification and is implemented by cross-entropy in this work.

Experiments

In this section, we conduct experiments on five datasets and seven baselines to demonstrate the privacy protection adaptability and the graph learning effectiveness of PoinDP based on a semi-supervised node classification task.

Dataset and Model Setup

Datasets. For datasets (see Appendix B.1), we chose three citation networks (Cora, Citeseer and PubMed) and two E-commerce networks in Amazon (Computers and Photo).

Baselines. For baselines, GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), and HyperIMBA (Fu et al. 2023) are convolutional neural networks model, attention neural networks model, and hierarchy-aware model, respectively. VANPD and LaP (Olatunji, Nejdl, and Khosla 2021) which use the Laplace noise perturbation mechanism are privacy models in Euclidean space. RdDP, AtDP, and the proposed PoinDP are privacy methods in hyperbolic spaces. RdDP and AtDP are two variant models of DP noise generation, representing the addition of random noise and attention-aware noise, respectively.

Model	Cora		Citeseer		PubMed		Computers		Photo	
	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1
GCN	80.0±1.1	80.1±1.1	68.1±0.2	68.6±0.2	78.5±0.5	78.5±0.5	84.7±2.3	82.5±3.6	90.2±1.4	89.6±1.6
GAT	81.6±1.1	81.8±1.0	69.4±1.2	70.0±1.0	77.0±0.5	77.0±0.4	87.5±0.4	87.1±0.5	**92.9±0.2**	**92.8±0.2**
HyperIMBA	**83.0±0.3**	**83.1±0.4**	**76.3±0.2**	**73.4±0.3**	**86.6±0.1**	**86.5±0.1**	**89.6±0.2**	**89.6±0.1**	92.8±0.3	92.5±0.3
VANPD	40.9±1.6	41.5±1.6	35.6±1.2	35.6±1.2	61.8±0.2	61.8±0.3	74.1±1.1	74.3±1.0	84.4±1.0	84.3±1.1
LaP	62.6±0.9	61.4±0.9	55.0±1.5	53.2±1.5	68.3±0.2	68.2±0.2	80.1±1.0	79.9±1.0	88.9±0.9	88.7±1.0
RdDP	78.1±0.2	75.1±0.4	73.1±0.5	70.0±0.7	79.1±0.7	78.6±0.9	80.5±0.9	76.1±1.6	91.4±0.2	90.1±0.5
AtDP	**81.0±0.2**	**80.0±0.2**	74.8±0.1	72.0±0.2	83.5±0.0	83.5±0.0	81.5±4.4	78.4±7.2	91.7±0.6	91.3±0.7
PoinDP	78.2±0.6	75.5±1.2	**75.5±0.2**	**72.5±0.2**	**83.8±0.2**	**83.7±0.2**	**86.9±0.4**	**86.5±0.5**	**92.6±0.2**	**92.4±0.3**

Table 1: Weighted-F1 and Micro-F1 score of the node classification task. (Result: average score ± standard deviation; Bold: the best of baseline model; Underline: runner-up.)

Model	Cora		Citeseer		PubMed		Computers		Photo	
	W-F1	Δ (%)	W-F1	Δ (%)	W-F1	Δ (%)	W-F1	Δ (%)	W-F1	Δ (%)
PoinDP	**59.9±1.4**	-	**74.0±1.3**	-	**79.4±0.5**	-	**83.8±0.4**	-	**91.9±0.5**	-
PoinDP (*w/o inter*)	48.4±2.6	↓11.5	60.6±4.1	↓13.4	70.8±1.4	↓8.6	79.5±1.8	↓4.3	91.3±0.4	↓0.6
PoinDP (*w/o intra*)	48.8±0.4	↓11.1	60.1±9.9	↓13.9	76.6±1.4	↓2.8	82.6±1.2	↓1.2	91.3±0.2	↓0.6
PoinDP (*w/o allocate*)	51.2±2.4	↓8.7	69.5±2.8	↓4.5	77.2±0.7	↓2.2	82.7±0.4	↓1.1	91.5±0.4	↓0.4

Table 2: Weighted-F1 scores (% ± standard deviation) and improvements (%) results of Ablation Study. (Result: average score ± standard deviation; Bold: best.)

Settings. PoinDP performs the semi-supervised node classification task to verify its privacy performance. Our dataset split follows the PyTorch Geometric. The learning rate lr is 0.005, the privacy budget ϵ to be $[0, 1]$, and the training iterations E to be 200. For other model settings, we adopt the default optimal values in the corresponding papers. We conducted the experiments with NVIDIA GeForce RTX 3090 with 16GB of Memory.

Performance Evaluation

Performance of Node Classification. We evaluate PoinDP for node classification where privacy models are trained in $\epsilon = 1$. The Weighted-F1 and Micro-F1 scores are reported in Table 1 where the best results are shown in bold and the runner-up results are shown in underline. It can be observed from the results that differential privacy-based models perform worse on the classification task compared with non-DP models while increasing the protection for sensitive information, which is caused by adding extra noise. Notably, PoinDP gets the absolute upper hand in terms of performance among privacy-preserving models compared to other privacy-preserving models. Because the hyperbolic noise is more adapted to the operations in the hierarchical structure, the destructive power in the Euclidean noise is significantly attenuated, resulting in uniformly higher performance. In conclusion, on the premise of improving the ability for privacy protection, PoinDP preserves the data availability as much as possible and improves the performance of the node classification task.

Ablation Study. In this subsection, we conduct the ablation study for PoinDP to validate the model utility provided by our consideration of node hierarchies (*w/o inter*) and correlations (*w/o intra*) on a hierarchical structure, and to remove the adaptive privacy budget allocation (*w/o allocate*) to these two properties, i.e., the optimization of hyperbolic noise is removed. We set $\epsilon = 0.01$ for easy observation. The results as shown in Table 2, indicate that missing any component of PoinDP leads to a degradation of the performance, where PoinDP (*w/o allocate*) has the smallest impact in most of the datasets, but numerically demonstrates the effectiveness of the privacy budget allocation. In addition, the one-sided perturbations in both PoinDP (*w/o inter*) and PoinDP (*w/o intra*) experiments reflect a strong influence on the model performance, suggesting that they have individualized perturbation rules for the nodes.

Case Study and Analysis of Sensitivity. *As a case study*, to verify the effectiveness of PoinDP in privacy protection and its generalization ability to hierarchical structures, three splits for Cora are provided as *Cora* (random sampling training set), *Top-level* and *Bottom-level* (sampling the top 33% and bottom 33% of the training samples, respectively), and their training sets with moderate, weak, and strong sensitivity, respectively. Note that the nodes are ordered from highest to lowest according to the Poincaré weights, indicating that the nodes range in sensitivity from lowest to highest. As shown in Fig. 3, Cora, which randomly samples the training nodes, has the best overall performance and is comparable to the Top-level, and finally Bottom-level.

For the analysis of sensitivity, we evaluate the model performance by setting different ϵ from 0.01 to 1, where ϵ measures the strength of the model's privacy protection, with smaller values indicating greater privacy protection power,

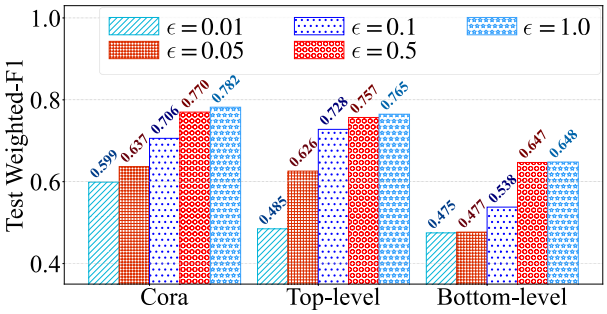


Figure 3: Hierarchical sensitivity experiments on Cora.

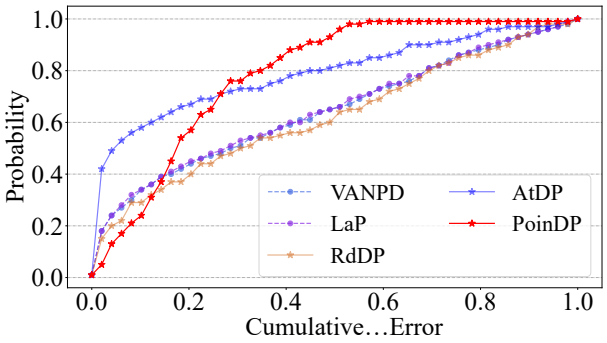
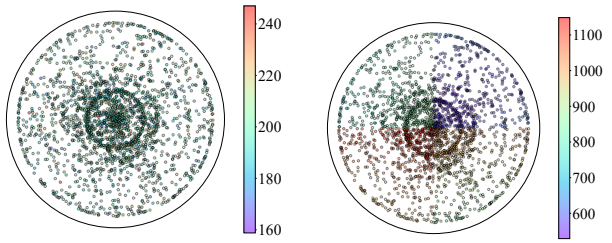


Figure 4: Cumulative error distribution with differential privacy-preserving method on Cora.



(a) VANPD. (b) PoinDP.

Figure 5: Visualization of noise distribution on Poincaré disk for VANPD and PoinDP on Cora.

Dataset / **Hyperbolicity** δ		**PubMed** $\delta = 1.65$		**Photo** $\delta = 0.15$	
	Metric	AUC ↑	Prec. ↑	AUC ↑	Prec. ↑
Attack	GCN	62.8±1.5	63.8±1.4	79.7±2.5	80.7±2.6
	GAT	58.4±2.4	59.3±2.8	79.7±0.9	80.4±0.8
	GCN+$\mathcal{H}$	**63.2±0.2**	**64.0±0.2**	**82.4±0.4**	**83.4±0.4**
	Metric	AUC ↓	Prec. ↓	AUC ↓	Prec. ↓
Defense	VANPD	51.1±0.3	51.2±0.5	68.1±1.6	68.9±1.5
	LaP	51.9±0.5	52.8±0.5	70.3±0.1	71.0±0.1
	PoinDP	**46.7±2.1**	**46.5±3.0**	**37.4±0.6**	**34.8±2.0**

Table 3: Membership Inference Attack (MIA) performance. (↑: the higher, the better; ↓: the lower, the better)

less usability, and more information loss. As shown in Fig. 3, for the overly strict $\epsilon = 0.01$, both the Top-level and the Bottom-level show the worst performance that can be understood, but in the looser limits, the Top-level samples perform well (these nodes are decisive for the downstream task so the amount of perturbation is low and the performance is almost close to Cora's). Whereas PoinDP in the Bottom-level samples adapts the requirement of needing a high degree of privacy preservation while providing the protection ability in a high privacy budget for sensitive data.

Analysis of Noise Distribution. Fig. 4 shows that the noise mechanism of PoinDP by the cumulative error distribution. We compare five privacy-preserving models and find that the error accumulation for PoinDP grows the fastest and ends its accumulation at 0.5, indicating a focused imposition of noise, and reflecting the individualized hierarchical perturbation mechanism of PoinDP. However, others are slow to converge, indicating a high percentage of results with large error values, and they aimlessly put noise into the samples, leading to poor usability. Overall, our hyperbolic Gaussian mechanism can put noise for some samples in a focused manner, providing personalized protection capability.

Visualization. We visualize the noise distribution of the four privacy models on the Cora dataset to intuitively represent the ability of our models to perceive hierarchical structures. Please refer to Appendix B.2 for other visualizations. Fig. 5 expresses the data as its whole with a hierarchical structure, where the colors represent the amount of noise and the position of each point is the layout of the node on the poincaré disk. As can be noticed in PoinDP in Fig. 5 (b), as the radius of the disk increases, the noise nodes become lighter in color and exhibit different colors at different angles, which fully demonstrates PoinDP's excellent ability to capture inter- and intra-hierarchy information. In contrast, the other models exhibit uniform perturbations to the hierarchy. In a nutshell, benefiting from the PHS and HGM mechanisms, PoinDP again demonstrates its effectiveness.

Attack Experiment. We conduct Membership Inference Attack (MIA) (Olatunji, Nejdl, and Khosla 2021) and the results are reported in Table 3. Please refer to Appendix B.3 for the detailed attack settings and performance analysis. The conclusion is that hierarchical information $\mathcal{H}$ can enhance the attacker's reasoning ability, while PoinDP can provide superior protective capabilities.

Conclusion

In this paper, for the first time, we propose the privacy leakage problem caused by the hierarchical structure of the graph and define the problem from the perspective of hyperbolic geometry. We propose PoinDP, a novel and unified privacy-preserving graph learning framework for the hierarchical privacy leakage issue. PoinDP designs personalized hierarchy-aware sensitivities and defines differential privacy techniques in hyperbolic space, and obtains high-quality graph representations while satisfying privacy guarantees. Experimental results empirically demonstrate the superior hierarchy perception capability of our framework and obtain excellent privacy preservation.

Acknowledgments

The corresponding authors are Xingcheng Fu and Chunming Hu. This paper is supported by the National Key Research and Development Program of China Grant (No. 2022YFB4501901) and the National Natural Science Foundation of China (No. U21A20474 and 62302023). We owe sincere thanks to all authors for their valuable efforts and contributions.

References

Bachmann, G.; Bécigneul, G.; and Ganea, O. 2020. Constant curvature graph convolutional networks. In *ICML*, 486–496. PMLR.

Chami, I.; Ying, Z.; Ré, C.; and Leskovec, J. 2019. Hyperbolic Graph Convolutional Neural Networks. In *NeurIPS*, 4869–4880.

Dwork, C. 2006. Differential Privacy. In *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, 1–12.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. D. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, 265–284.

Dwork, C.; and Roth, A. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4).

Fu, X.; Wei, Y.; Sun, Q.; Yuan, H.; Wu, J.; Peng, H.; and Li, J. 2023. Hyperbolic Geometric Graph Representation Learning for Hierarchy-imbalance Node Classification. In *WWW*, 460–468. ACM.

Ganea, O.; Bécigneul, G.; and Hofmann, T. 2018. Hyperbolic Neural Networks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *NeurIPS*, 5350–5360.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*, 1024–1034.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Krioukov, D.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; and Boguná, M. 2010. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3): 036106.

Li, J.; Fu, X.; Zhu, S.; Peng, H.; Wang, S.; Sun, Q.; Yu, P. S.; and He, L. 2023. A Robust and Generalized Framework for Adversarial Graph Embedding. *IEEE Trans. Knowl. Data Eng.*, 11004–11018.

Lin, W.; Li, B.; and Wang, C. 2022. Towards Private Learning on Decentralized Graphs With Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.*, 17: 2936–2946.

Liu, Q.; Nickel, M.; and Kiela, D. 2019. Hyperbolic Graph Neural Networks. In *NeurIPS*, 8228–8239.

Nagano, Y.; Yamaguchi, S.; Fujita, Y.; and Koyama, M. 2019. A Wrapped Normal Distribution on Hyperbolic Space for Gradient-Based Learning. In *ICML*, volume 97, 4693–4702. PMLR.

Nickel, M.; and Kiela, D. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*, 6338–6347.

Olatunji, I. E.; Nejdl, W.; and Khosla, M. 2021. Membership Inference Attack on Graph Neural Networks. In *3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*, 11–20. IEEE.

Papadopoulos, F.; Kitsak, M.; Serrano, M. Á.; Boguná, M.; and Krioukov, D. 2012. Popularity versus similarity in growing networks. *Nature*, 537–540.

Qin, Z.; Yu, T.; Yang, Y.; Khalil, I.; Xiao, X.; and Ren, K. 2017. Generating Synthetic Decentralized Social Graphs with Local Differential Privacy. In *ACM SIGSAC*, 425–438. ACM.

Ren, J.; Jiang, L.; Peng, H.; Lyu, L.; Liu, Z.; Chen, C.; Wu, J.; Bai, X.; and Yu, P. S. 2022. Cross-Network Social User Embedding with Hybrid Differential Privacy Guarantees. In *CIKM*, 1685–1695.

Sajadmanesh, S.; and Gatica-Perez, D. 2021. Locally Private Graph Neural Networks. In *CCS*, 2130–2145. ACM.

Sajadmanesh, S.; Shamsabadi, A. S.; Bellet, A.; and Gatica-Perez, D. 2023. Gap: Differentially private graph neural networks with aggregation perturbation. In *USENIX*.

Sala, F.; De Sa, C.; Gu, A.; and Ré, C. 2018. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 4460–4469. PMLR.

Sun, L.; Wang, F.; Ye, J.; Peng, H.; and Yu, P. S. 2023a. CONGREGATE: Contrastive Graph Clustering in Curvature Spaces. In *IJCAI*, 2296–2305.

Sun, L.; Ye, J.; Peng, H.; Wang, F.; and Yu, P. S. 2023b. Self-Supervised Continual Graph Learning in Adaptive Riemannian Spaces. In *AAAI*, 4633–4642.

Sun, L.; Ye, J.; Peng, H.; and Yu, P. S. 2022a. A Self-supervised Riemannian GNN with Time Varying Curvature for Temporal Graph Learning. In *CIKM*, 1827–1836.

Sun, L.; Zhang, Z.; Ye, J.; Peng, H.; Zhang, J.; Su, S.; and Yu, P. S. 2022b. A Self-Supervised Mixed-Curvature Graph Neural Network. In *AAAI*, 4146–4155.

Sun, L.; Zhang, Z.; Zhang, J.; Wang, F.; Peng, H.; Su, S.; and Yu, P. S. 2021a. Hyperbolic Variational Graph Neural Network for Modeling Dynamic Graphs. In *AAAI*.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Ji, C.; and Yu, P. S. 2022c. Graph Structure Learning with Variational Information Bottleneck. In *AAAI*, 4165–4174.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Ning, Y.; Yu, P. S.; and He, L. 2021b. SUGAR: Subgraph Neural Network with Reinforcement Pooling and Self-Supervised Mutual Information Mechanism. In *WWW*, 2081–2091.

Sun, Q.; Li, J.; Yuan, H.; Fu, X.; Peng, H.; Ji, C.; Li, Q.; and Yu, P. S. 2022d. Position-aware Structure Learning for Graph Topology-imbalance by Relieving Under-reaching and Over-squashing. In *CIKM*.

Tifrea, A.; Bécigneul, G.; and Ganea, O. 2019. Poincare Glove: Hyperbolic Word Embeddings. In *ICLR*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wei, C.; Ji, S.; Liu, C.; Chen, W.; and Wang, T. 2020. AsgLDP: Collecting and Generating Decentralized Attributed Graphs With Local Differential Privacy. *IEEE TIFS*, 15: 3239–3254.

Wei, Y.; Fu, X.; Sun, Q.; Peng, H.; Wu, J.; Wang, J.; and Li, X. 2022. Heterogeneous Graph Neural Network for Privacy-Preserving Recommendation. In *ICDM*.

Wu, Z.; Zhan, M.; Zhang, H.; Luo, Q.; and Tang, K. 2022. MTGCN: A multi-task approach for node classification and link prediction in graph data. *Inf. Process. Manag.*, 59(3): 102902.

Yang, C.; Wang, H.; Zhang, K.; Chen, L.; and Sun, L. 2021. Secure Deep Graph Generation with Link Differential Privacy. In *IJCAI*, 3271–3278.

Ye, Q.; Hu, H.; Au, M. H.; Meng, X.; and Xiao, X. 2022. LF-GDPR: A Framework for Estimating Graph Metrics With Local Differential Privacy. *IEEE TKDE*, 34(10): 4905–4920.

Yuan, H.; Sun, Q.; Fu, X.; Zhang, Z.; Ji, C.; Peng, H.; and Li, J. 2024. Environment-Aware Dynamic Graph Learning for Out-of-Distribution Generalization. *NeurIPS*, 36.

Zhang, G.; Cheng, D.; Yuan, G.; and Zhang, S. 2024. Learning fair representations via rebalancing graph structure. *Inf. Process. Manag.*, 61(1): 103570.

Zhang, Z.; Chen, M.; Backes, M.; Shen, Y.; and Zhang, Y. 2022. Inference Attacks Against Graph Neural Networks. In Butler, K. R. B.; and Thomas, K., eds., *USENIX*, 4543–4560.