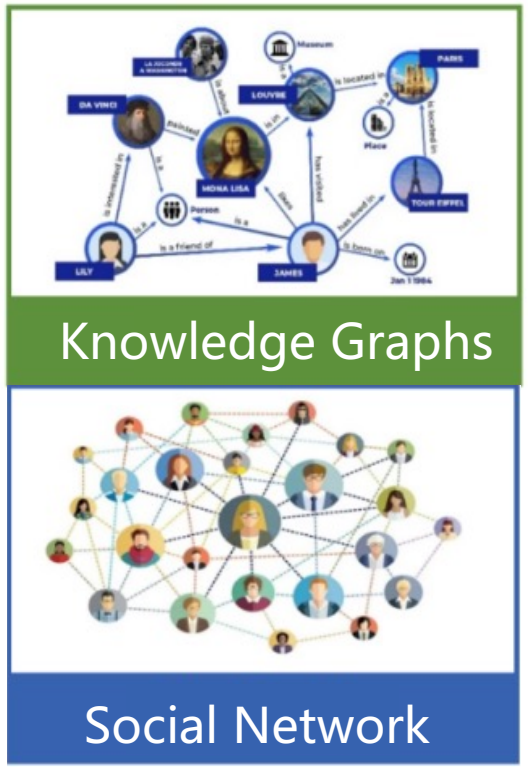
Poincaré Differential Privacy for Hierarchy-aware Graph Emebedding

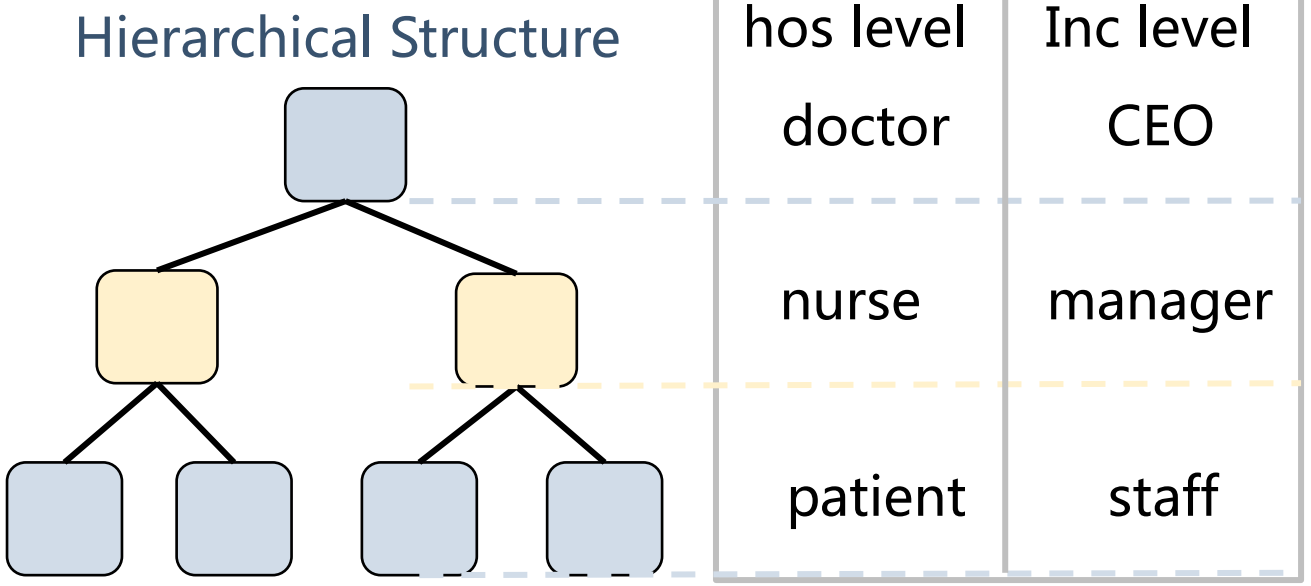
Anonymous submission

2023.10.16

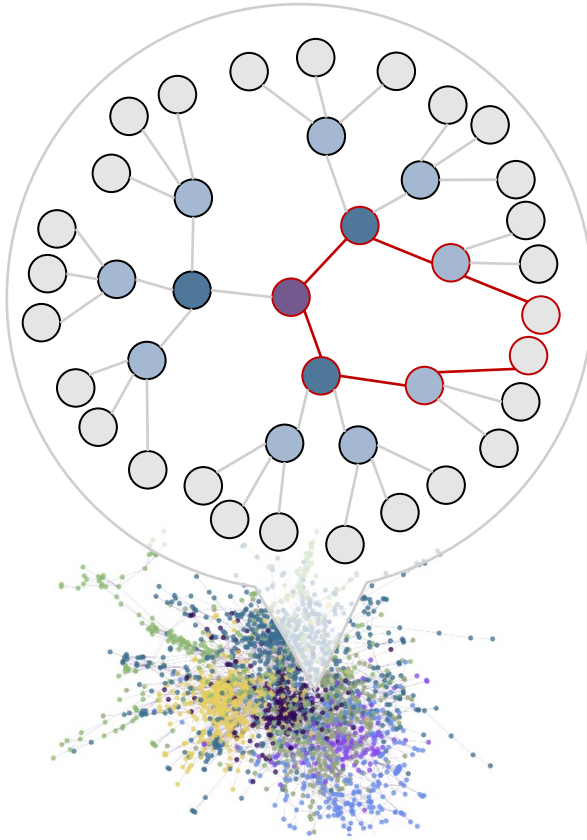
Hierarchy V.S. Graph Neural Networks



(a) Graph Data

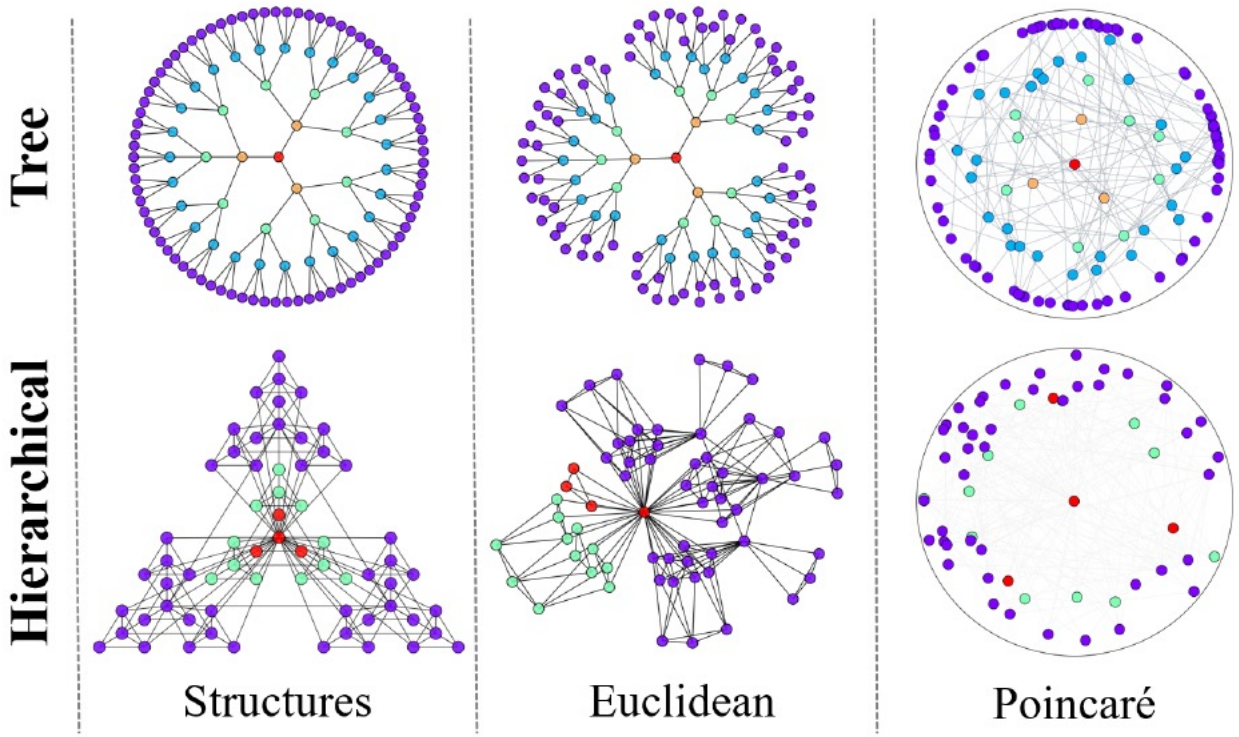


(b) Hierarchical Info.

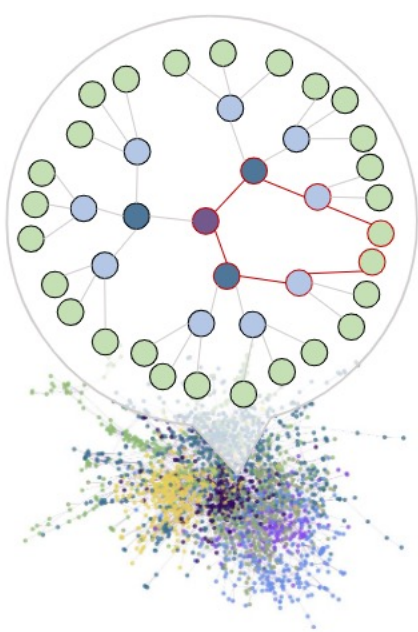


(c) Node embeddings

Privacy leakage on the hierarchical structure



(b) Tree, hierarchical structure in hyperbolic space.



(a) Latent Tree Structure.

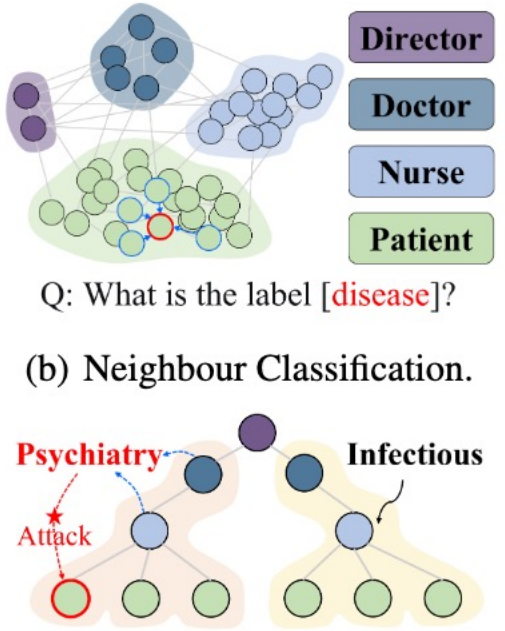
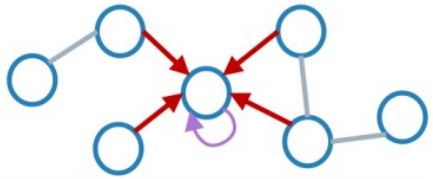


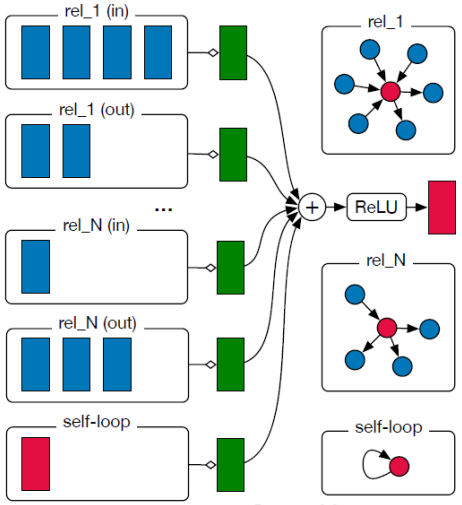
Figure 1: Privacy leakage on the hierarchical structure.

- Hierarchy can be better captured by Hyper-GNNs, but leads to privacy leackage!
- Out-of-distribution (OOD) generalized GNNs are critically needed!

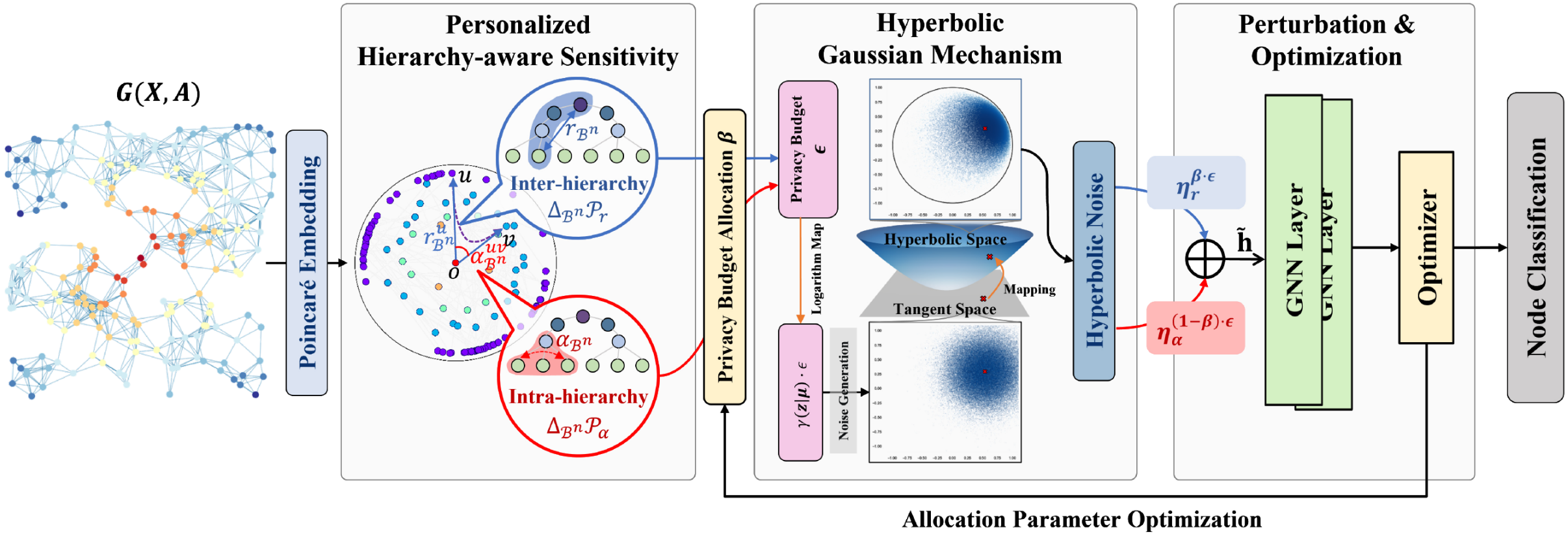
Main Challenges

- How to use geometric priors of hyperbolic space to perceive the hierarchical structure of a graph, ensuring that sensitive information is not subjected to inference attacks
- Traditional privacy protection methods only consider the privacy of neighbors or relationships, but weak perception of the hierarchical structure of data.
- Hierarchy information can be captured in hyperbolic space, while existing privacy protection techniques can only measure the privacy of nodes in Euclidean space and do not have the ability to measure in hyperbolic space.

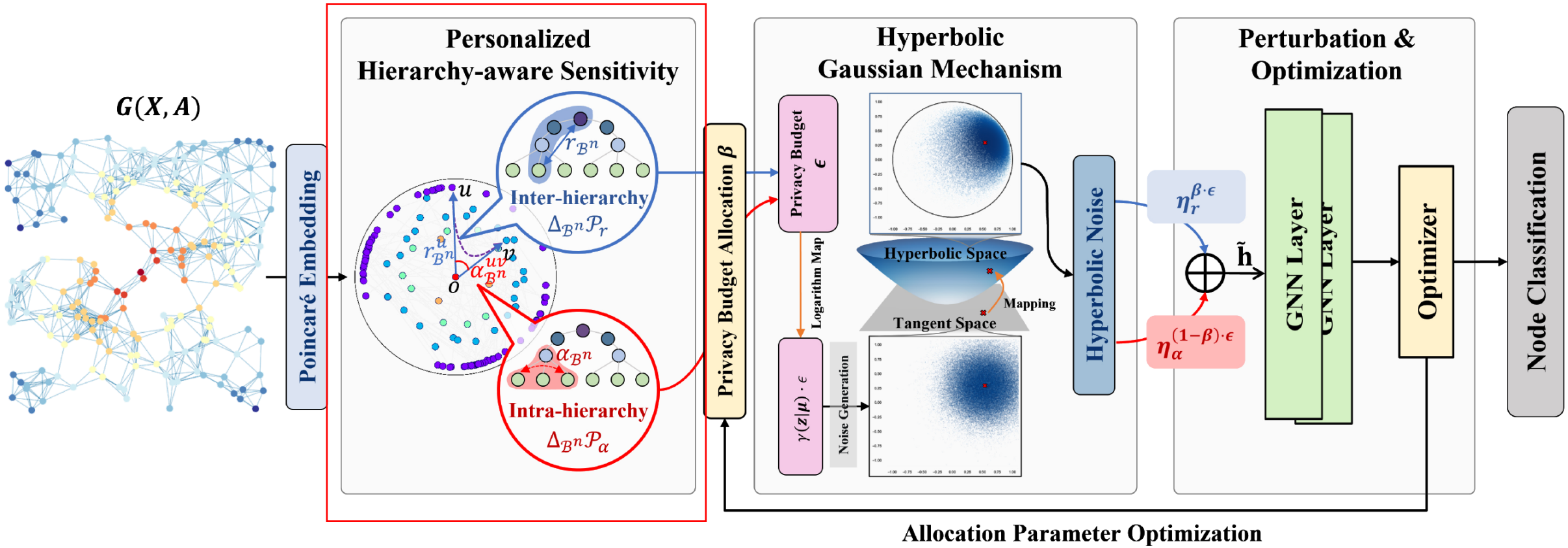




Framework


G(X, A)
Poincaré Embedding
Personalized Hierarchy-aware Sensitivity
Inter-hierarchy
Intra-hierarchy
Privacy Budget Allocation β
Hyperbolic Gaussian Mechanism
Privacy Budget ϵ
Logarithm Map
Noise Generation
Hyperbolic Space
Mapping
Tangent Space
Hyperbolic Noise
Perturbation & Optimization
GNN Layer
Optimizer
Node Classification
Allocation Parameter Optimization

Framework



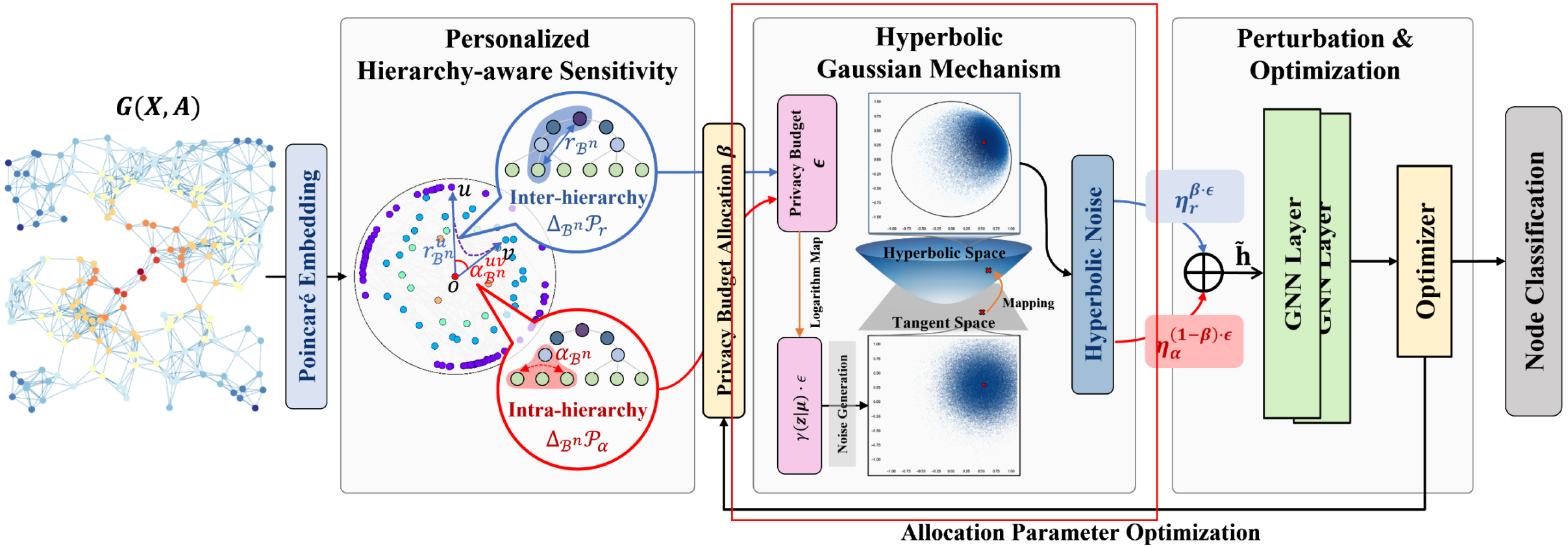
Definition 6 (Inter-hierarchy Sensitivity) *Given V and V' are the neighboring subsets of graph nodes, and V and V' only differ by one node. The inter-hierarchy sensitivity can be defined as:*

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

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:*

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

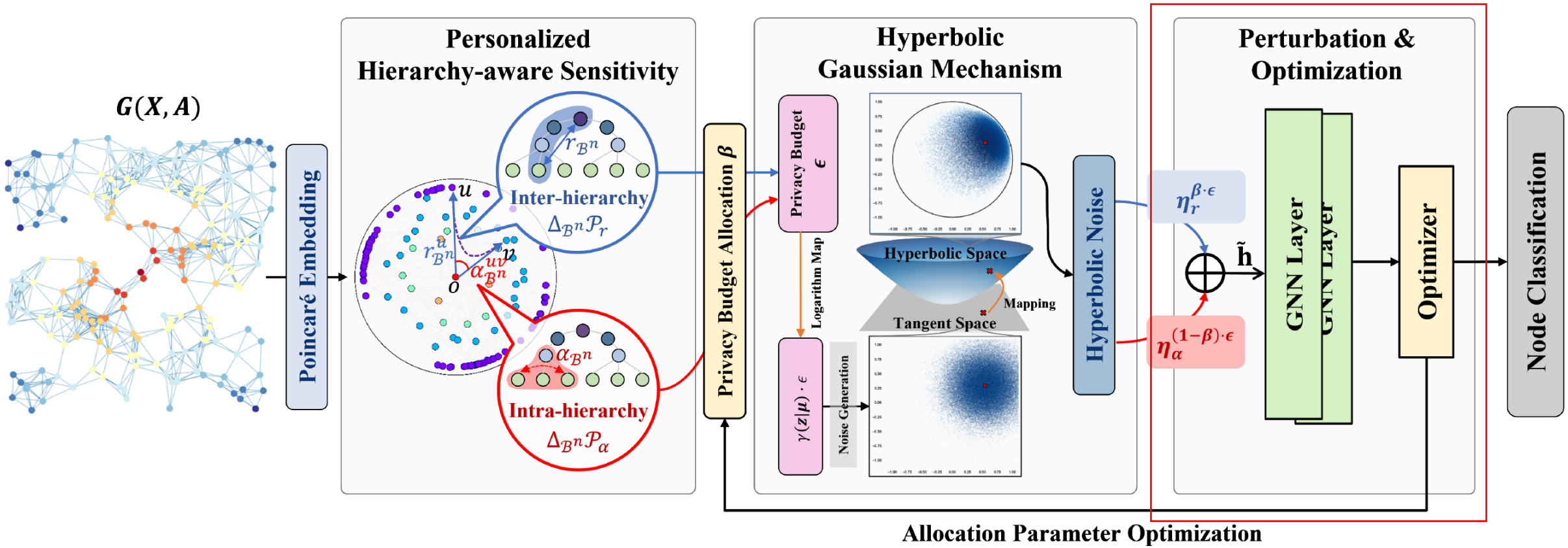
Framework



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.*

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

Framework



$$\mathbf{h}_u^{(l+1)} = \text{SOFTMAX}\left(\sum_{v\in\mathcal{V}(u)} c_v \mathbf{W}^{(l)}\mathbf{h}_v^{(l)}\right), \quad \hat{\mathbf{h}} = \mathbf{h} + \eta_r^{\beta\cdot\epsilon} + \eta_\alpha^{(1-\beta)\cdot\epsilon}, \quad \mathcal{L} = \frac{1}{\|\mathcal{V}_U\|}\sum_{v\in\mathcal{V}_U}\mathcal{L}_\mathcal{G}(\hat{\mathbf{h}}_{u,v}, y_v),$$

Framework

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 $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \nabla\mathbf{\Theta}$.
end

Node Classification

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.)

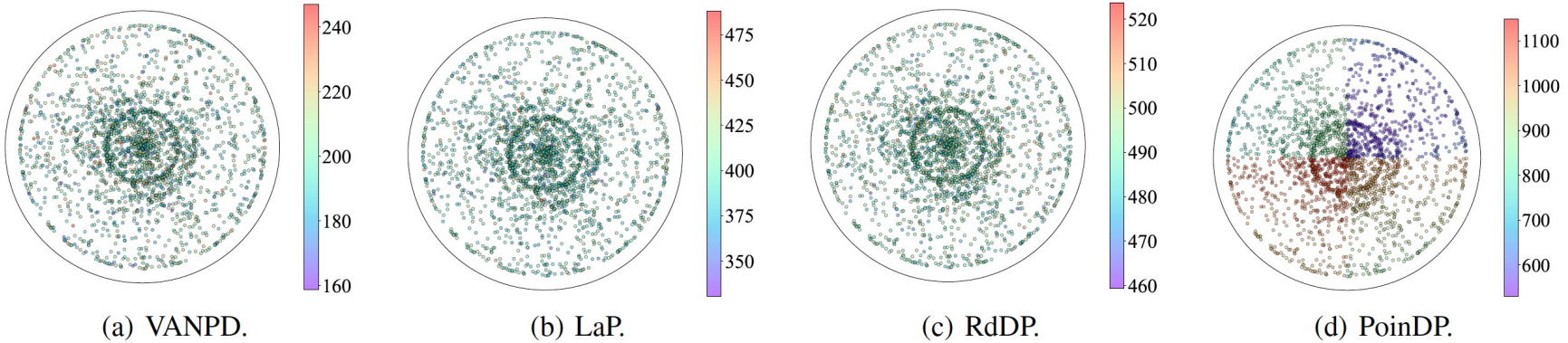
Model	Cora		Citeseer		PubMed		Computers		Photo	
	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1
GCN	80.0±1.1	80.1±1.1	68.1±0.2	68.6±0.2	78.5±0.5	78.5±0.5	84.7±2.3	82.5±3.6	90.2±1.4	89.6±1.6
GAT	81.6±1.1	81.8±1.0	69.4±1.2	70.0±1.0	77.0±0.5	77.0±0.4	87.5±0.4	87.1±0.5	**92.9±0.2**	**92.8±0.2**
HyperIMBA	**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**

Ablation Study

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

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

Visualization



(a) VANPD. (b) LaP. (c) RdDP. (d) PoinDP.

Figure 3: Visualization of noise distribution on Poincaré disk for four privacy models on Cora.

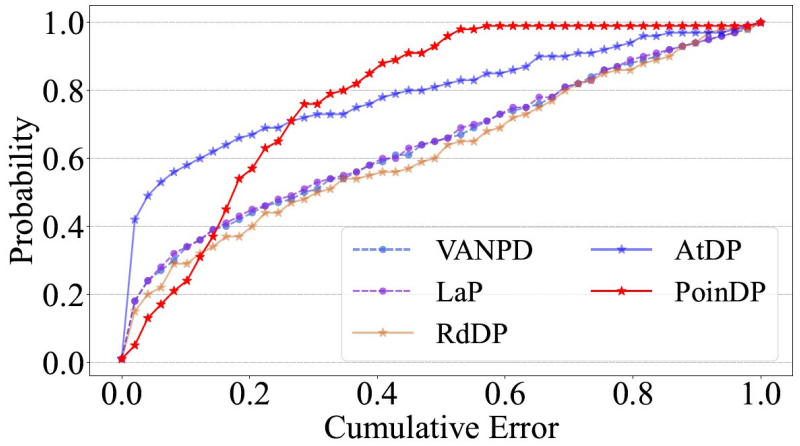


Figure 4: Cumulative error distribution with differential privacy-preserving method on Cora.

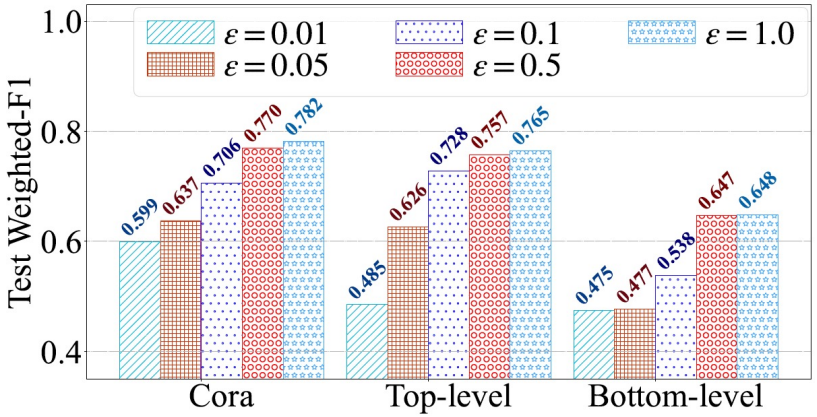


Figure 5: Hierarchical sensitivity experiments on Cora.

Conclusions

- 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.
- The Personalized Hierarchy-aware Sensitivity can measure the sensitivity of the hierarchical structure and adaptively allocate the privacy protection strength. We extend the Gaussian mechanism to hyperbolic space to realize random perturbations that satisfy differential privacy 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

Poincaré Differential Privacy for Hierarchy-aware Graph Emebedding

Anonymous submission

2023.10.16