Privacy-Preserving Machine Learning on Graphs

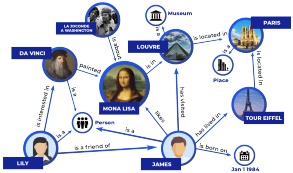
Sina Sajadmanesh Idiap Research Institute, EPFL

594 Special Topics: Socially Responsible AI: Theories and Practice
University of Illinois at Chicago
October 2022

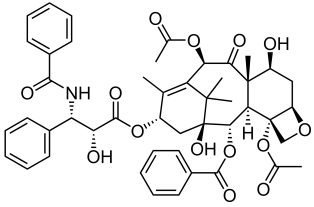
Outline

Introduction

Graphs are Ubiquitous



Knowledge Graphs



Molecules

Social Networks

Image source (from left to right): https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/, https://en.wikipedia.org/wiki/Terpenoid, https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

Node Classification / Regression

- Given a graph, which is the class label / value of a node?
- **Example:** face account detection

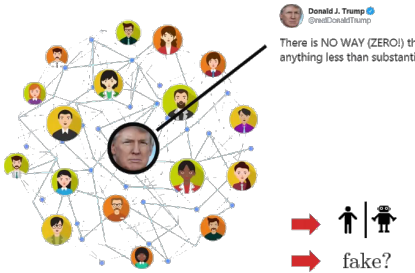


Image Source: "A gentle introduction to graph neural networks" by Andreas Loukas.

Link Prediction

- Given a graph, which links are likely to form?
- **Example:** recommendation systems

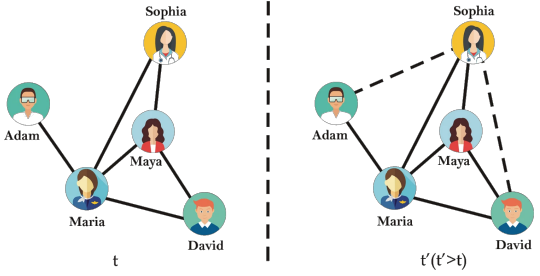
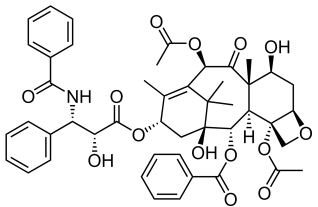


Image source: [Ahmad et al., 2020]

Graph Classification

- Given a graph, predict its label
- Example: antibiotic discovery

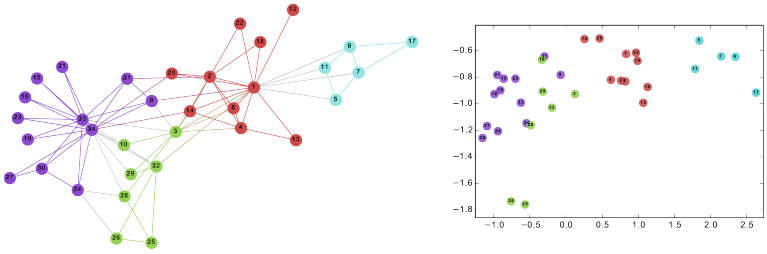


Antibiotic? Or Not

Image source: https://en.wikipedia.org/wiki/Terpenoid

Graph Representation Learning

- We need to learn representation of nodes in a low-dimensional space
  - Similar nodes in the graph should be mapped close to each other in the embedding space



- **Graph Neural Networks** (GNNs) are **state-of-the-art** representation learning algorithms for graphs.

Image source: [Perozzi et al., 2014]

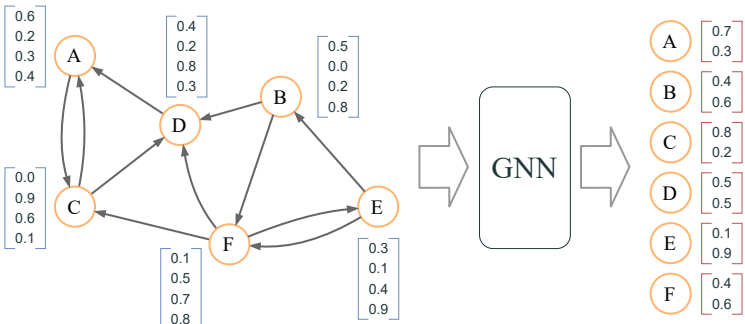
Privacy Concerns

- Graph data could be privacy-sensitive
  - e.g., users' personal attributes, financial transactions, medical/biological networks, ...
- Graph-based ML algorithms are vulnerable to privacy attacks
  - e.g., **link stealing attack** [He et al., 2021] or **membership inference attack** [Olatunji et al., 2021]

We need privacy-preserving machine learning algorithms for graph data!

Graph Neural Networks

GNNs Learn Node Embeddings



Internal Structure of GNNs

$\mathbf{A}$: Adjacency matrix

$\mathbf{X}$: Input node features

$\mathbf{Y}$: Predicted node labels

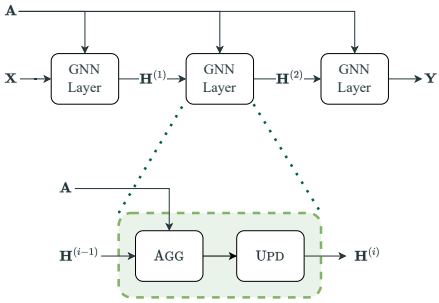
$\mathbf{H}^{(i)}$: Hidden node representations of layer i

AGG: Aggregation function

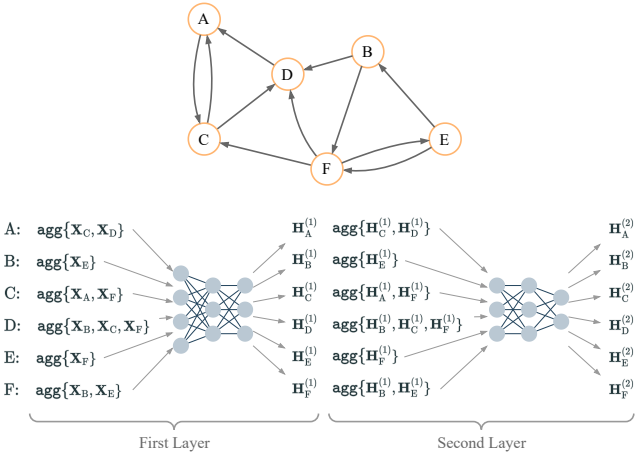
- e.g., summation: $\text{AGG}(\mathbf{H}, \mathbf{A}) = \mathbf{A}^T \cdot \mathbf{H}$

UPD: Learnable update function

- e.g., an MLP



GNNs Unfolded



Privacy Attacks on GNNs

Membership Inference: Key Idea

- Exploiting the statistical difference of the posterior class probabilities between train and test nodes
  - GNNs are more confident when predicting labels for the training data
  - Nodes with high output confidence are likely members of the training set

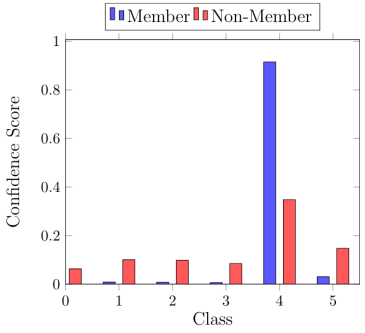
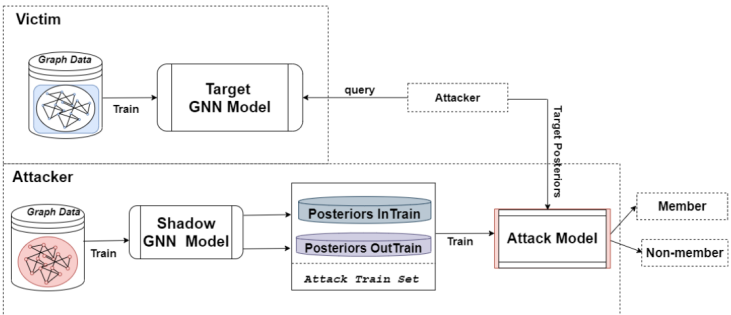
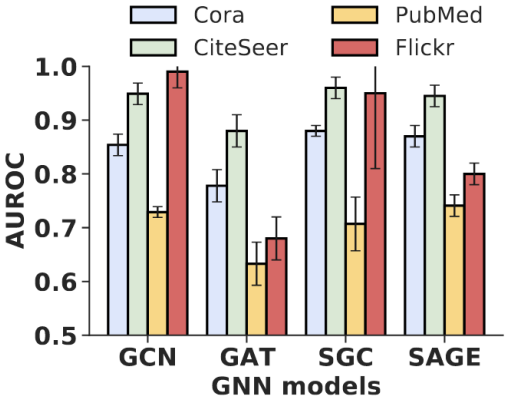


Image source: [Duddu et al., 2020]

- Assumptions:
  - Attacker has access to the posterior class probabilities of the GNN
  - Attacker has access to a shadow graph dataset similar to the target graph
- Attack Methodology [Olatunji et al., 2021]:



Membership Inference: Attack Results



[Olatunji et al., 2021]

Link Inference: Idea and Assumptions

Exploits the similarity of prediction posteriors for connected nodes

- If two nodes are connected, then their prediction scores are likely similar

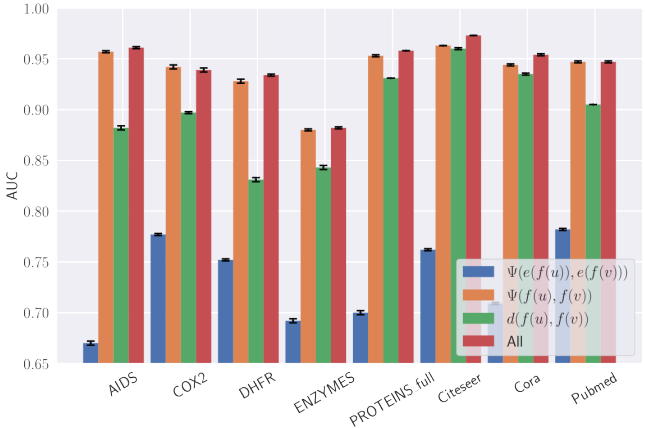
Assumptions

- Attacker has access to the posterior class probabilities of the GNN
- Attacker has access to an auxiliary subgraph of the original graph

Link Inference: Attack Methodology [He et al., 2021]

- Obtain the prediction scores from the target GNN for every node pair in the auxiliary graph
- Extract features from the obtained scores for each node pair
  - features based on **distance metrics** (cosine, euclidean, etc), **vector operations** (average, hadamard product, etc), and **entropy**
- Train an MLP using the extracted features and the link state in the auxiliary graph
- Use the trained MLP to infer the link between any node pair in the original graph

Link Inference: Attack Results



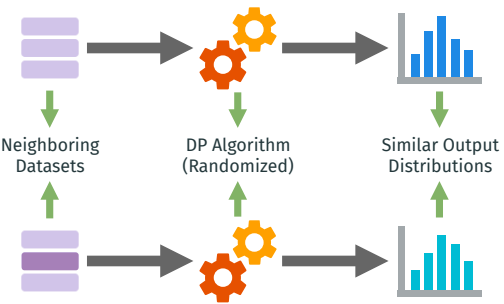
[He et al., 2021]

Differential Privacy

Differential Privacy [Dwork et al., 2006]

Randomized algorithm A is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs S:

$$\Pr[A(D) \in S] \leq e^{\epsilon} \Pr[A(D') \in S] + \delta$$



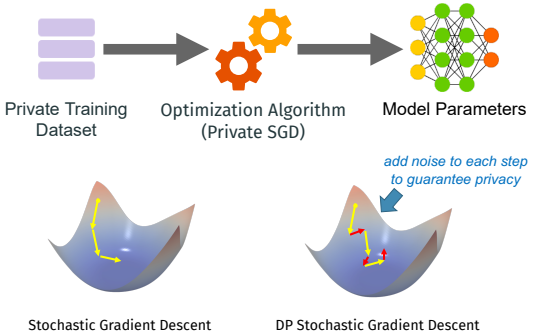
Differential Privacy [Dwork et al., 2006]

Randomized algorithm A is (ϵ, δ)-differentially private if for all neighboring datasets $D \simeq D'$ and all sets of outputs S:

$$\Pr[A(D) \in S] \leq e^{\epsilon} \Pr[A(D') \in S] + \delta$$

- The neighboring relation captures what is protected
  - **Standard DP:** D and D' differ by at most **one record**
  - **Edge-level DP:** D and D' are graphs differing by at most **one edge**
  - **Node-level DP:** D and D' are graphs differing by at most **one node** (and all its adjacent edges)

Differentially private learning is possible with noisy gradient descent



DP-SGD Algorithm [Abadi et al., 2016]

input : Data $\{\vec{x}_1 \ldots, \vec{x}_N\}$, learning rate η, batch size B, epochs T, clipping threshold C, noise variance σ^2,

Initialize $\vec{\theta}_0$ randomly

for $t \in [T \cdot \frac{N}{B}]$ **do**

  Sample a batch $\vec{B}_t$ by selecting each $\vec{x}_i$ independently with probability $\frac{B}{N}$

  For each $\vec{x}_i \in \vec{B}_t$: $\vec{g}_t(\vec{x}_i) \leftarrow \nabla_{\vec{\theta}_t} L(\vec{\theta}_t, \vec{x}_i)$ // compute per-sample gradients

  $\tilde{\vec{g}}_t(\vec{x}_i) \leftarrow \text{clip}(\vec{g}_t(\vec{x}_i), C)$ // clip gradients to max norm C

  $\tilde{\vec{g}}_t \leftarrow \frac{1}{B}\left(\sum_{\vec{x}_i \in \vec{B}_t} \tilde{\vec{g}}_t(\vec{x}_i) + \mathcal{N}(0, \sigma^2 \vec{I})\right)$ // add Gaussian noise with variance σ^2

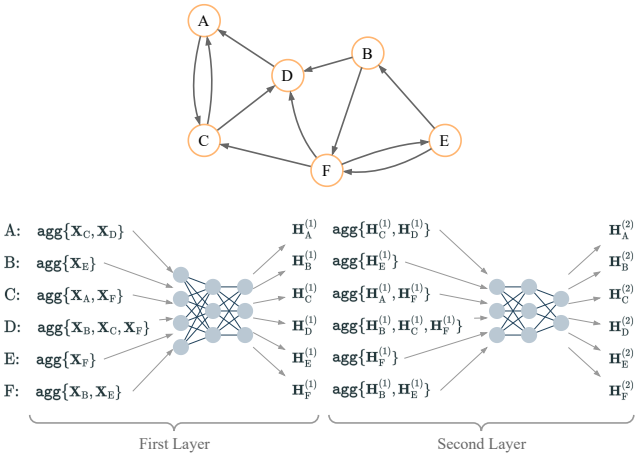
  $\vec{\theta}_{t+1} \leftarrow \vec{\theta}_t - \eta \tilde{\vec{g}}_t$ // SGD step

end

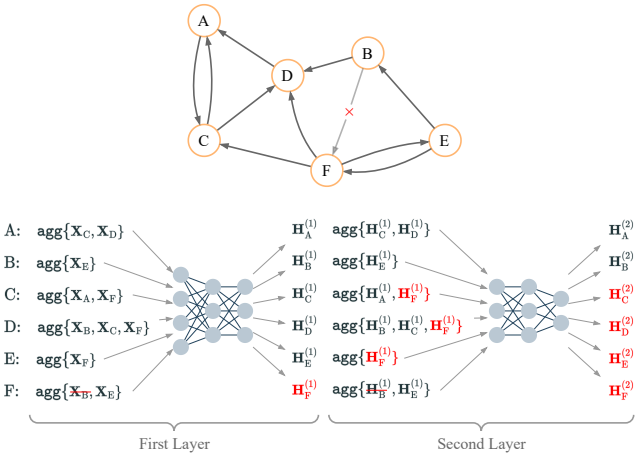
output: $\vec{\theta}_{\frac{TN}{B}}$

Differentially Private GNNs

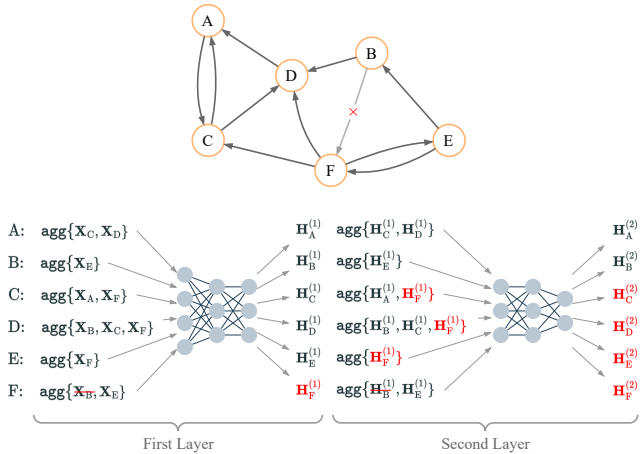
DP GNN Challenges: Exploding Sensitivity



DP GNN Challenges: Exploding Sensitivity

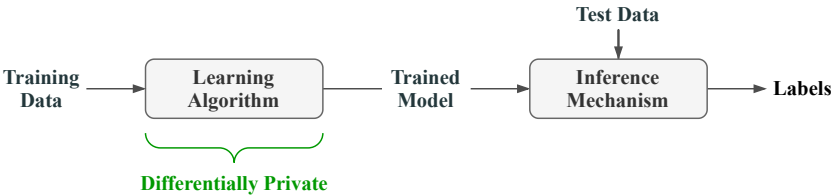


DP GNN Challenges: Exploding Sensitivity



The number of affected outputs = $\mathcal{O}(\textit{max degree}^{\textit{num layers}})$

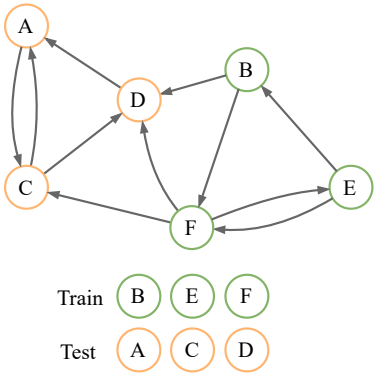
Private Learning: Standard Neural Nets



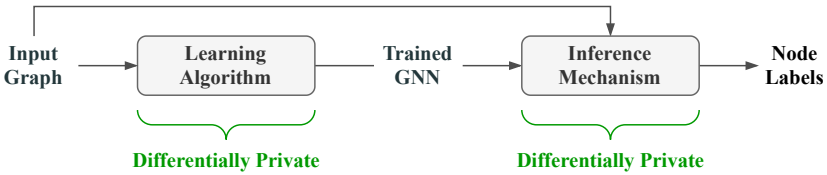
Inference is independent of the training data

Differentially Private GNN Challenges: Inference Privacy

- GNN re-uses graph data for inference
- Private information leaks at inference, even with a private model



Private Learning: Graph Neural Nets



Both training and inference should be private

Our Approach: Aggregation Perturbation

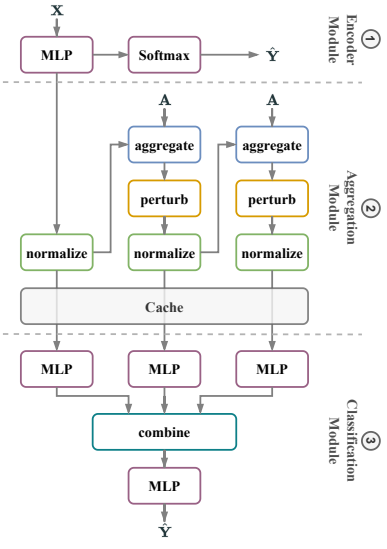
- **Aggregation Perturbation:** adding noise to output of the aggregation step
  - Prevents the exploding sensitivity problem by composing differentially private aggregation steps
  - Ensures inference privacy
- Applying aggregation perturbation to the conventional GNNs is costly
  - Every forward pass of the model consumes privacy budget
  - The excessive noise results in poor performance

Our Approach: Aggregation Perturbation

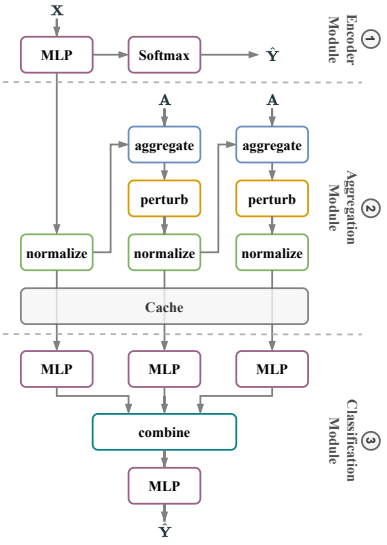
- **Aggregation Perturbation:** adding noise to output of the aggregation step
  - Prevents the exploding sensitivity problem by composing differentially private aggregation steps
  - Ensures inference privacy
- Applying aggregation perturbation to the conventional GNNs is costly
  - Every forward pass of the model consumes privacy budget
  - The excessive noise results in poor performance

Need to tailor the GNN architecture to the private learning setting!

GNN with Aggregation Perturbation (GAP) [Sajadmanesh et al., 2022]

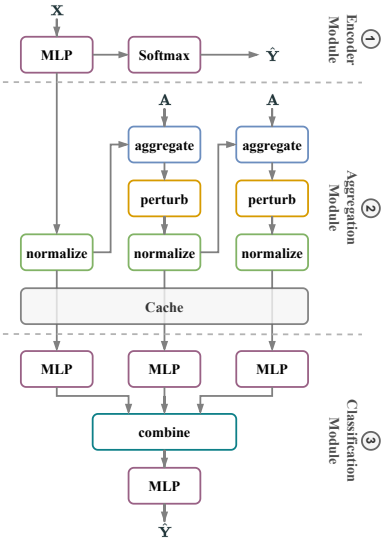


1. Encoder Module
   - Learns to encode node features into lower-dimensional representations
   - Does not use graph adjacency information



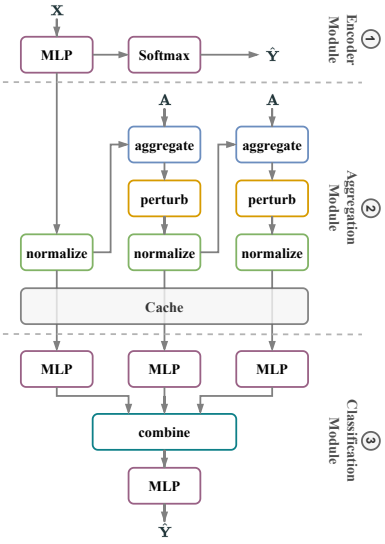
GNN with Aggregation Perturbation (GAP) [Sajadmanesh et al., 2022]

1. Encoder Module
   - Learns to encode node features into lower-dimensional representations
   - Does not use graph adjacency information
2. Aggregation Module
   - Computes aggregated node representations at multiple hops privately using the aggregation perturbation approach
   - Uses graph adjacency information



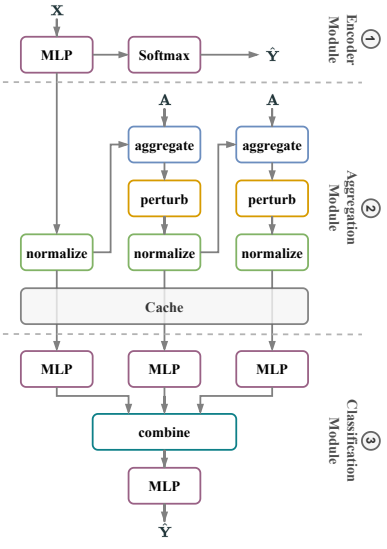
GNN with Aggregation Perturbation (GAP) [Sajadmanesh et al., 2022]

1. Encoder Module
   - Learns to encode node features into lower-dimensional representations
   - Does not use graph adjacency information
2. Aggregation Module
   - Computes aggregated node representations at multiple hops privately using the aggregation perturbation approach
   - Uses graph adjacency information
3. Classification Module
   - Learns to perform node-wise classification based on aggregated node representations
   - Does not re-use graph adjacency information



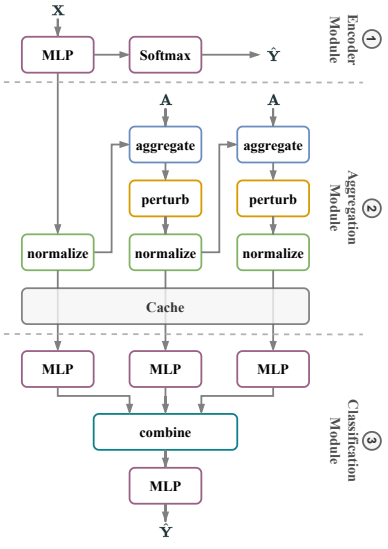
Advantages of GAP Architecture

✓ Edge-level DP



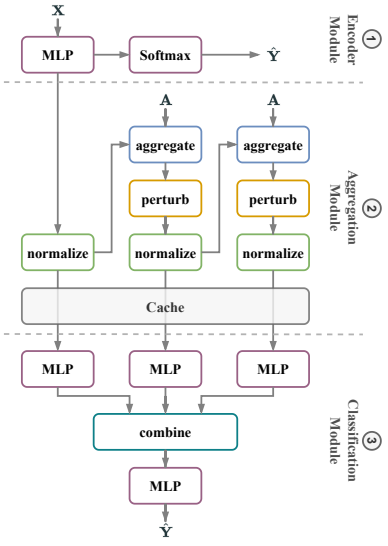
Advantages of GAP Architecture

- ✓ Edge-level DP
- ✓ Node-level DP through combination with DP-SGD
  - For bounded-degree graphs



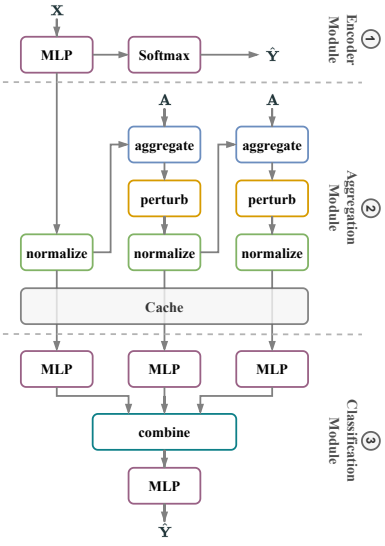
Advantages of GAP Architecture

- ✓ Edge-level DP
- ✓ Node-level DP through combination with DP-SGD
  - For bounded-degree graphs
- ✓ Multi-hop aggregations



Advantages of GAP Architecture

- ✓ Edge-level DP
- ✓ Node-level DP through combination with DP-SGD
  - For bounded-degree graphs
- ✓ Multi-hop aggregations
- ✓ Zero-cost inference privacy



Experiment Settings: Datasets

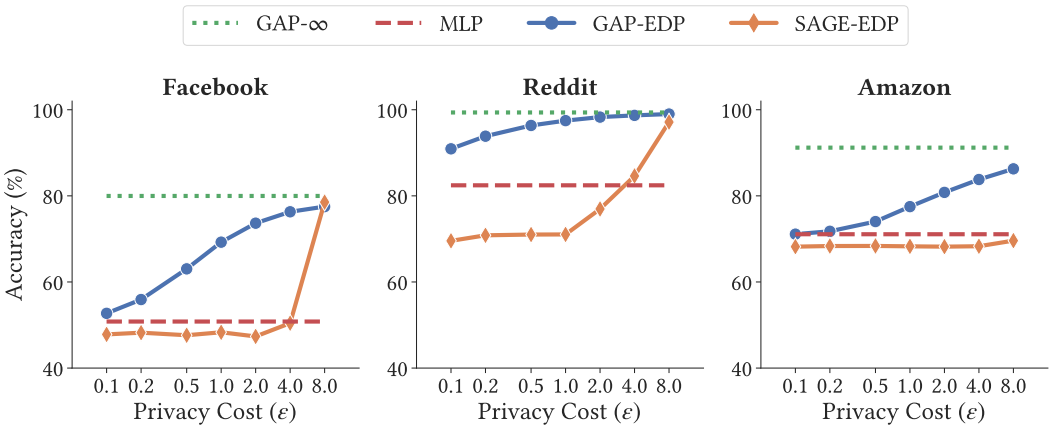
- **Task:** Node Classification

Dataset	Classes	Nodes	Edges	Features	Avg. Degree
Facebook	6 Year	26,406 User	2,117,924 Friendship	501	62
Reddit	8 Community	116,713 Post	46,233,380 Mutual User	602	209
Amazon	10 Category	1,790,731 Product	80,966,832 Mutual Purchase	100	22

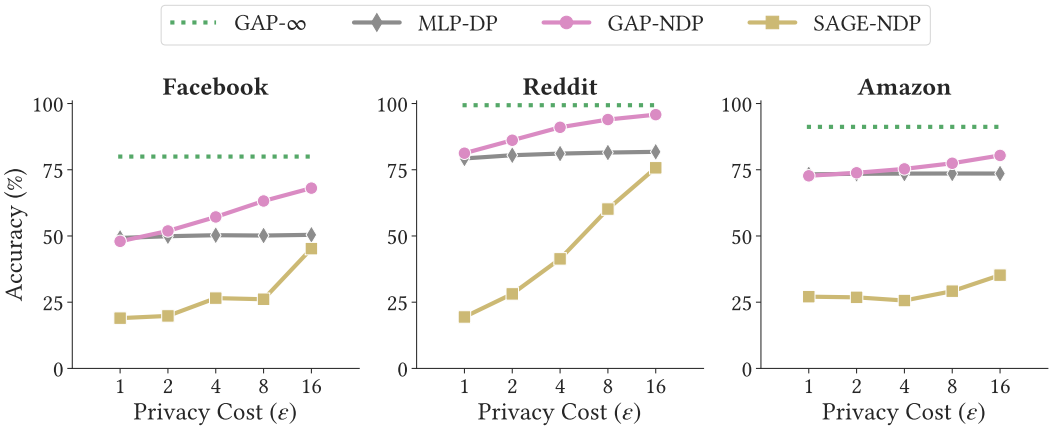
Accuracy of Non-Private Methods

Method	Facebook	Reddit	Amazon
GAP-∞	80.0 $\pm$ 0.48	**99.4 $\pm$ 0.02**	91.2 $\pm$ 0.07
SAGE-∞	**83.2 $\pm$ 0.68**	99.1 $\pm$ 0.01	**92.7 $\pm$ 0.09**

Edge-Level DP Accuracy-Privacy Trade-Off



Node-Level DP Accuracy-Privacy Trade-Off



Mean AUC of node-level membership inference attack.

Dataset	Method	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = \infty$
Facebook	GAP-NDP	50.16	50.25	50.61	51.11	52.66	81.67
	SAGE-NDP	50.25	50.20	50.23	50.17	50.20	62.49
	MLP-DP	50.32	50.72	52.13	53.44	54.77	81.57

Conclusion and Future Work

Conclusion

- GNNs leak private information
  - They are vulnerable to privacy attacks
- Implementing DP in GNNs is challenging
  - Exploding sensitivity
  - Inference privacy
- Our Differentially Private GNN: GAP
  - Ensures both edge-level and node-level DP
  - Supports multi-hop aggregations
  - Provides inference privacy

Open Problems

- How to achieve DP in more expressive GNN architectures?
- How to achieve DP in link-level or graph-level tasks?
- How to achieve DP in dynamically changing graphs?
- How to achieve DP in heterogeneous graphs (e.g., knowledge graphs)?

Thank You!

Questions?

sajadmanesh@idiap.ch

References I


Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016).
Deep learning with differential privacy.
In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.


Ahmad, I., Akhtar, M. U., Noor, S., and Shahnaz, A. (2020).
Missing link prediction using common neighbor and centrality based parameterized algorithm.
Scientific Reports, 10(1):1–9.

References II

Duddu, V., Boutet, A., and Shejwalkar, V. (2020).
Quantifying privacy leakage in graph embedding.
In *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '20, page 76–85, New York, NY, USA. Association for Computing Machinery.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
Calibrating noise to sensitivity in private data analysis.
In *Theory of cryptography conference*, pages 265–284. Springer.

He, X., Jia, J., Backes, M., Gong, N. Z., and Zhang, Y. (2021).
Stealing links from graph neural networks.
In *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

References III

Olatunji, I. E., Nejdl, W., and Khosla, M. (2021).
Membership inference attack on graph neural networks.
arXiv preprint arXiv:2101.06570.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014).
Deepwalk: Online learning of social representations.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Sajadmanesh, S., Shamsabadi, A. S., Bellet, A., and Gatica-Perez, D. (2022).
Gap: Differentially private graph neural networks with aggregation perturbation.
arXiv preprint arXiv:2203.00949.