Privacy in GNNs

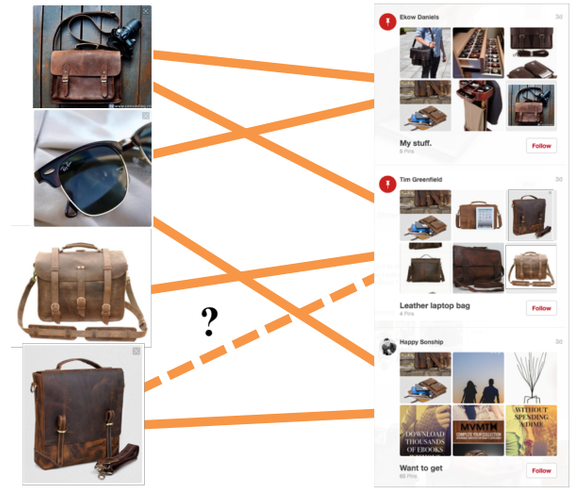
Megha Khosla

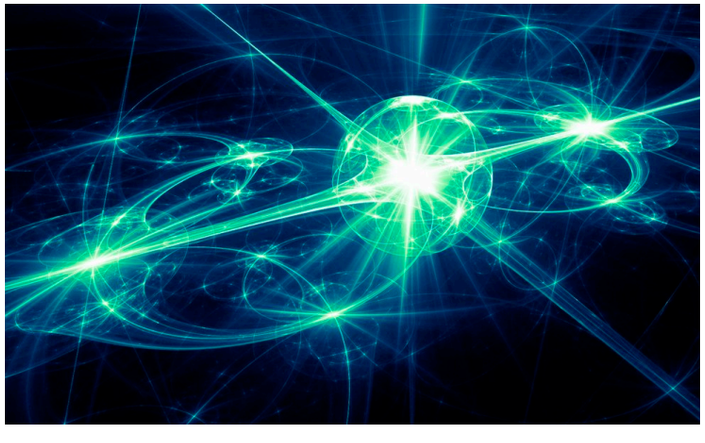
TU Delft

Success of Machine Learning for Graphs

discover **novel antibiotics** (Stokes *et al.*, Cell'20)

power **web-scale recommender systems**
(Ying *et al.*, KDD'18; Pal *et al.*, KDD'20)

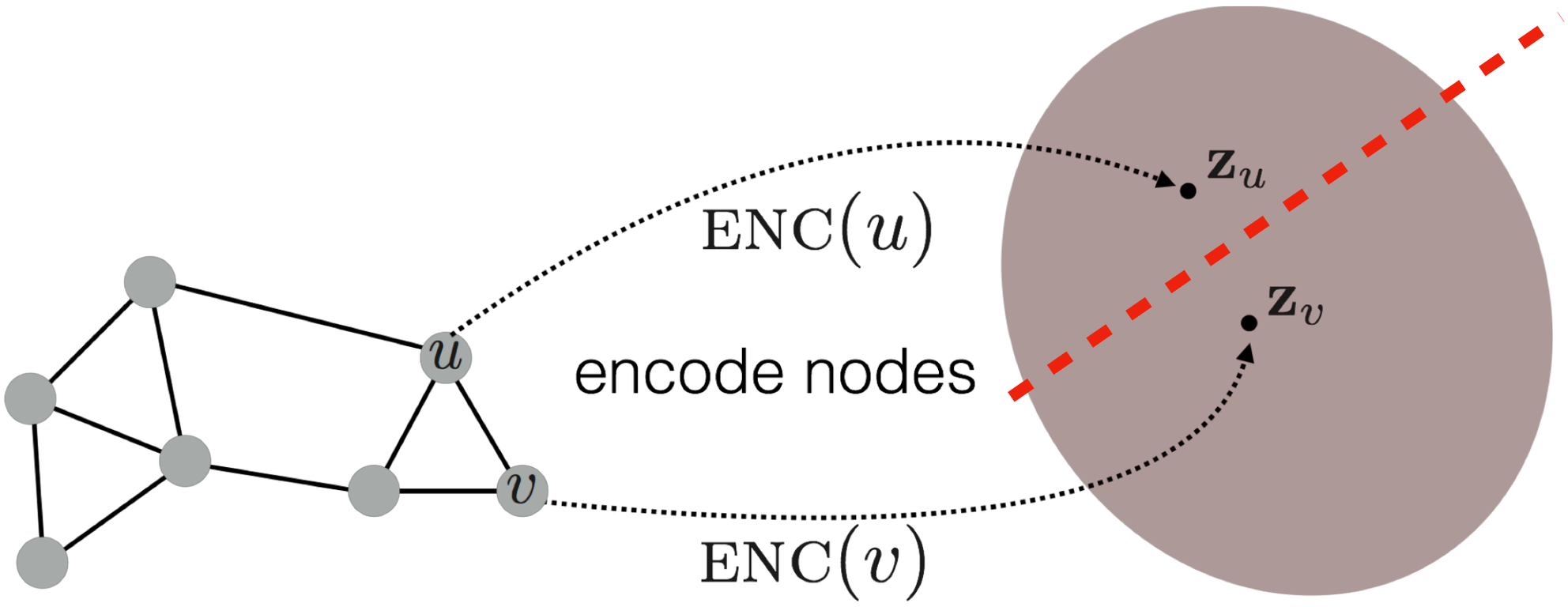




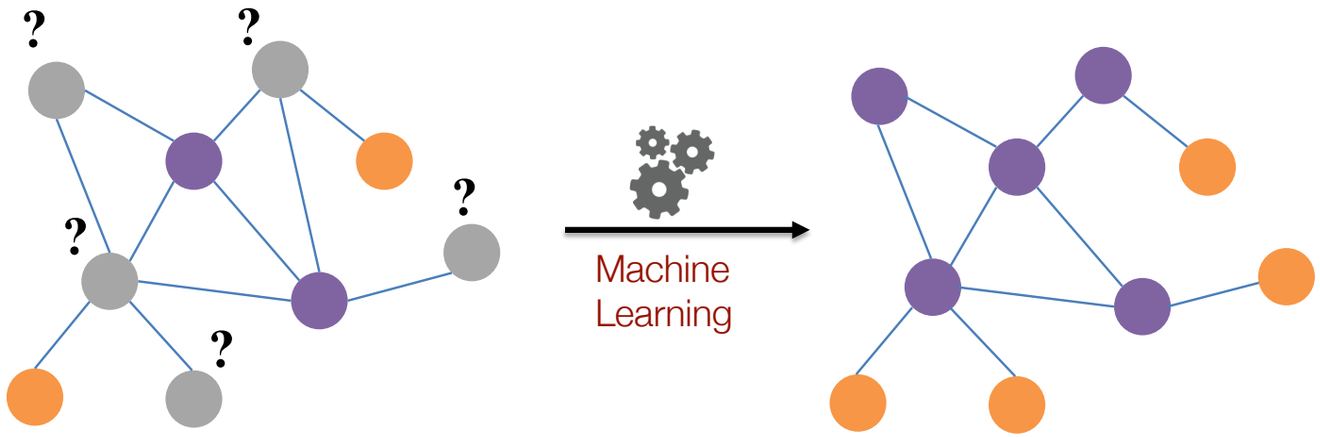
assist **particle physicists** (Martinez *et al.*, Eur. Phys. J. Plus'19)

TU Delft

Representation Learning



Machine Learning for Graphs



Node classification

Link prediction

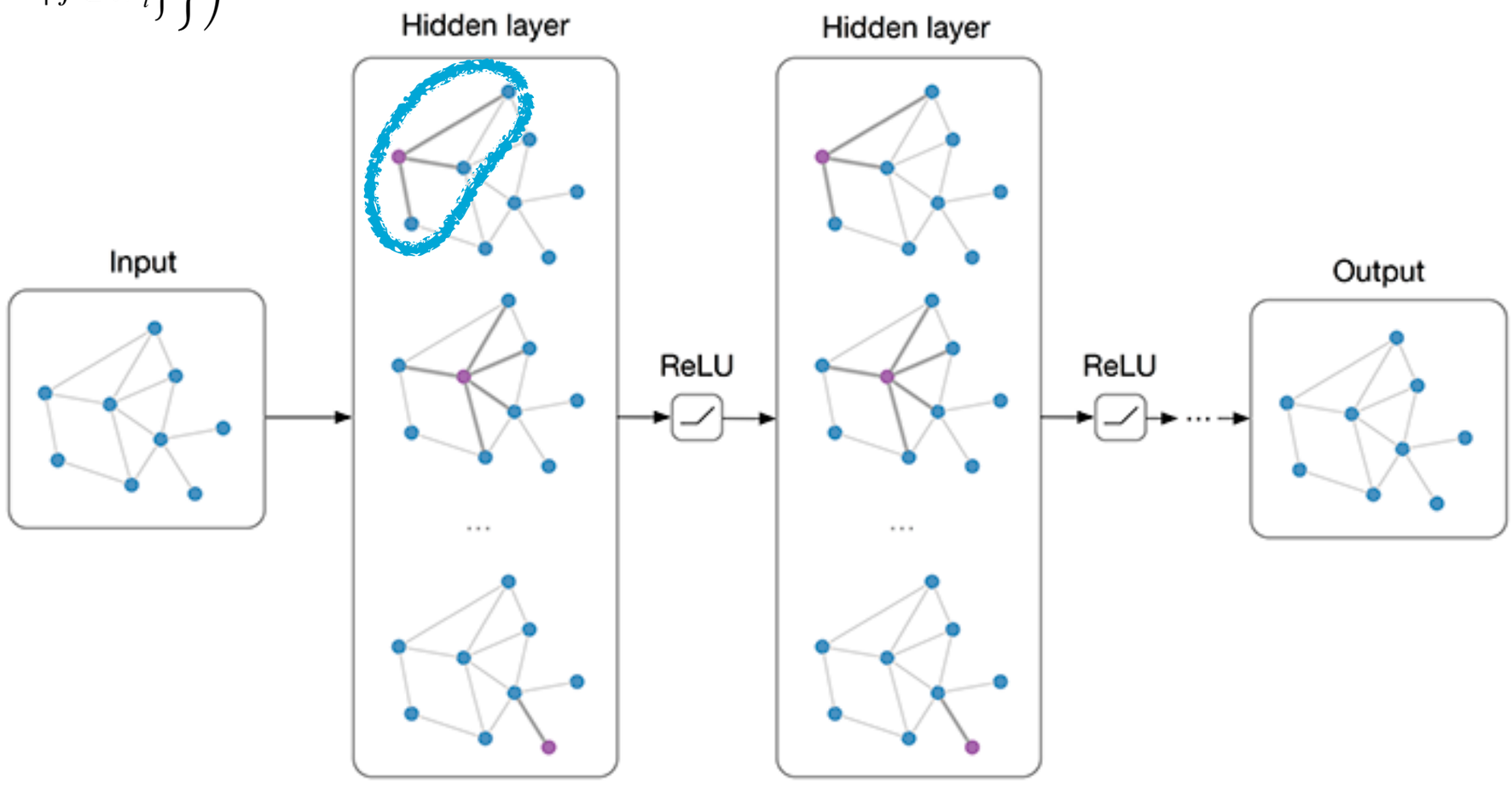
Graph classification

Community detection

Graph Neural Networks

$$z_i^{(\ell)} = \text{AGGREGATE}\left(\left\{x_i^{(\ell-1)}, \left\{x_j^{(\ell-1)} \mid j \in \mathcal{N}_i\right\}\right\}\right)$$

$$x_i^{(\ell)} = \text{TRANSFORM}\left(z_i^{(\ell)}\right)$$



Recursive aggregation over neighborhood feature representation

Image Source : https://tkipf.github.io/graph-convolutional-networks/

Privacy in GNNs

Graphs can contain sensitive information

- User's sensitive attributes
- Sensitive relations

GNNs encode relation information within the model, could memorise such information

- Your identity could be revealed because of your neighbour

Privacy in GNNs

Quantify Information Leakage in trained GNNs

- Node level inference attacks [Olatunji et al., ’21] [Duddu et al., ’20]
- Link level inference attacks [He et al., ’21] [Zhang et al., ’20]

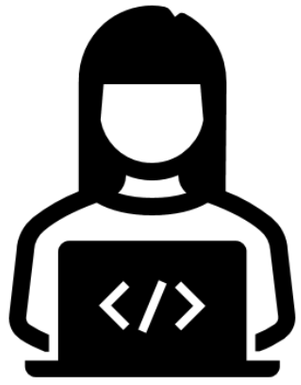
Privacy Preserving GNNs

- Centralised Setting [Olatunji et al., ’21]
- Federated Settings [Jian et al., ’22] [Sajadmanesh and Gatica-Perez, ’21]

Membership Inference Attack- Motivation

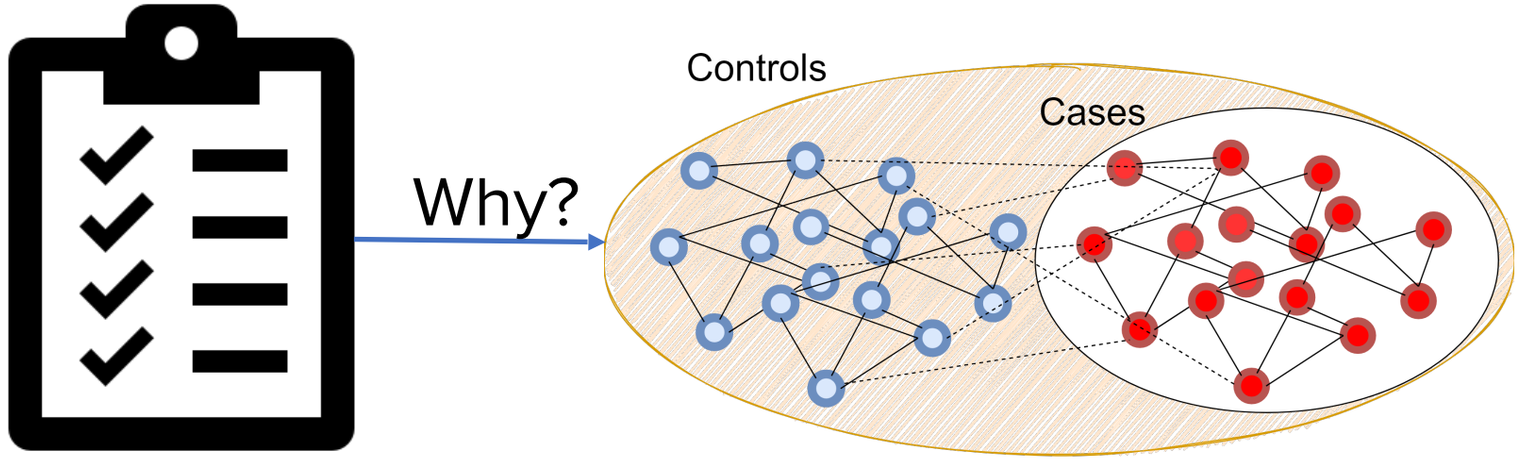
List of infected patients

Researcher

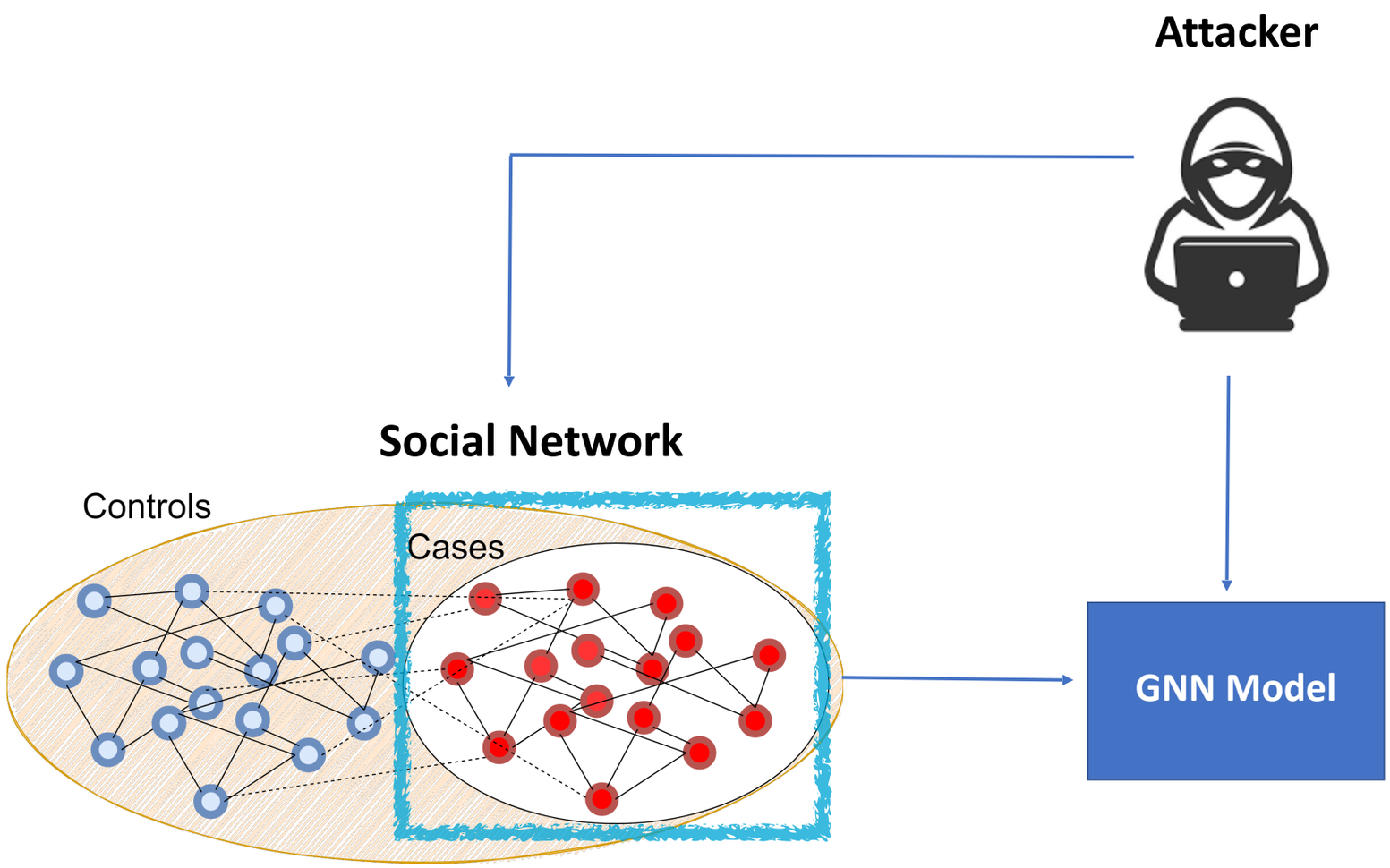


TU Delft

Membership Inference Attack- Motivation



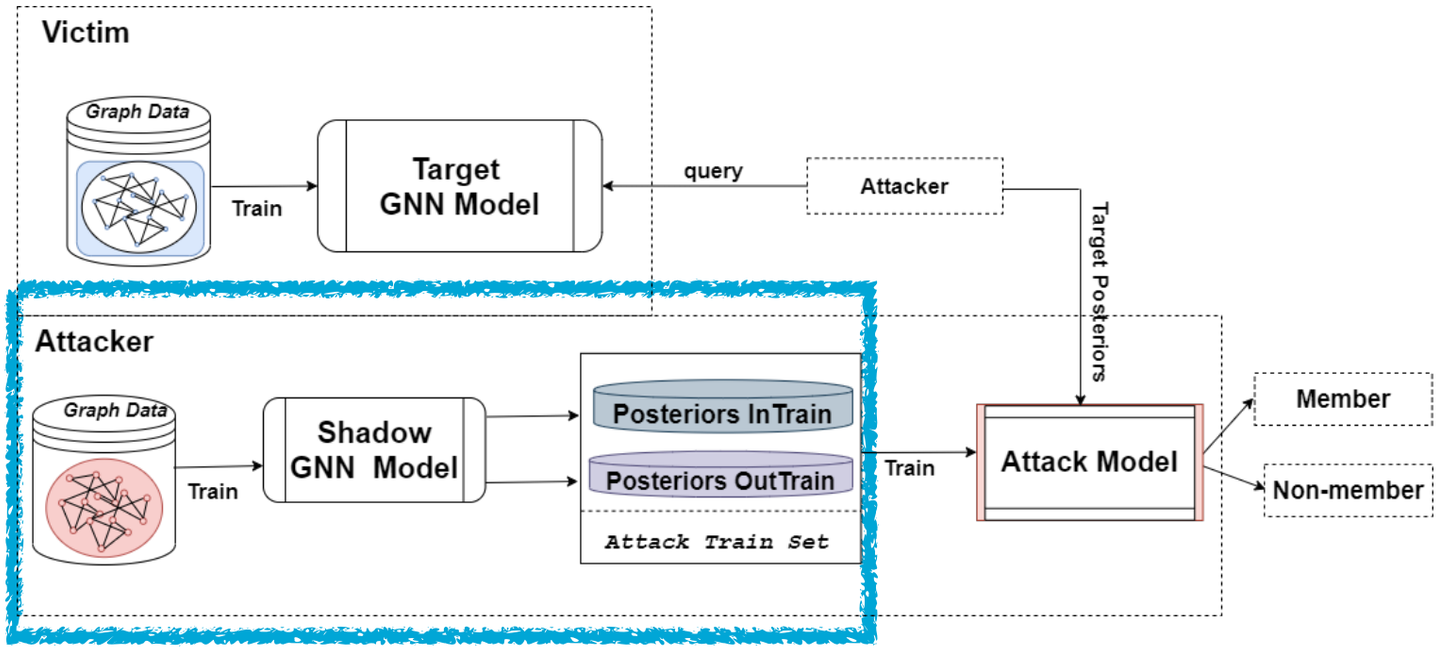
Membership Inference Attack



[Olatunji, Nejdl, Khosla, In IEEE TPS '21] **(Best student paper)**

https://arxiv.org/pdf/2101.06570.pdf

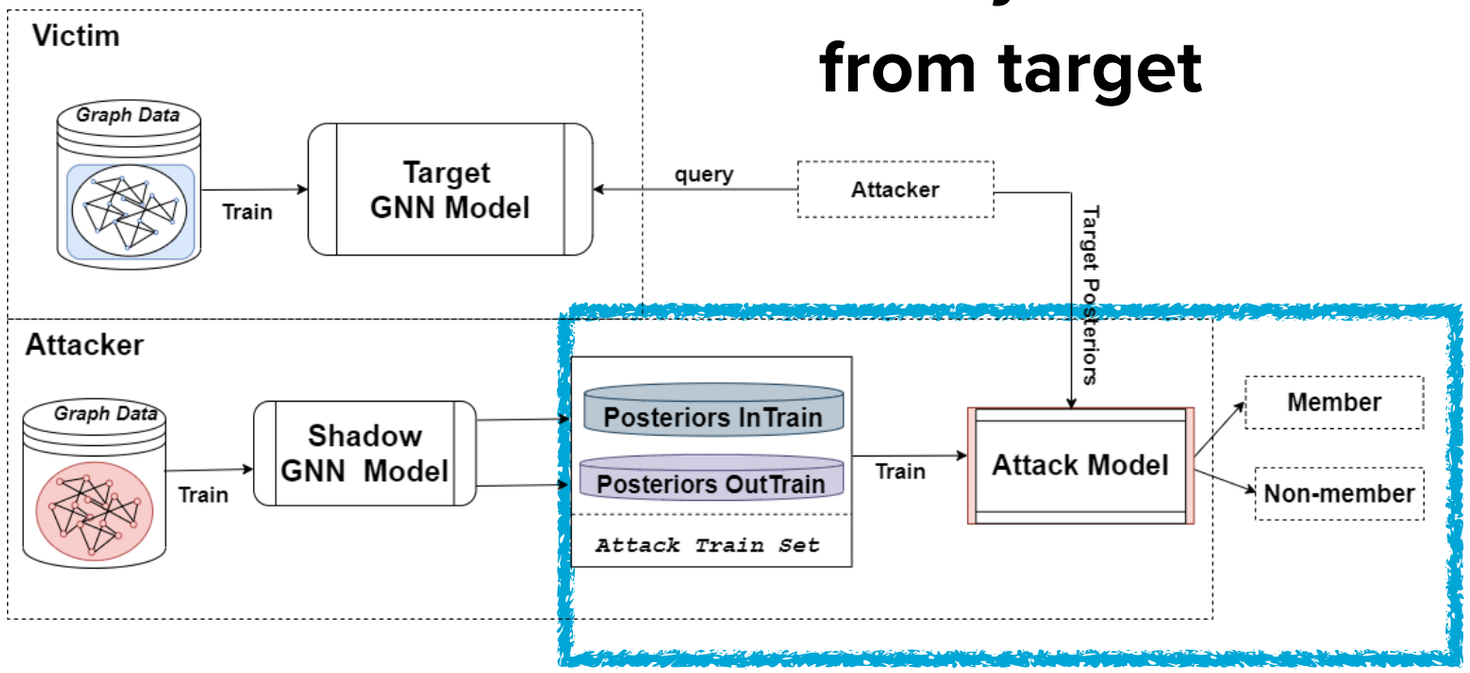
Attack Strategy



Train a shadow model to replicate behavior of target model

Assumption: Presence of a shadow dataset drawn out from the same distribution as the target (could be relaxed, see paper)

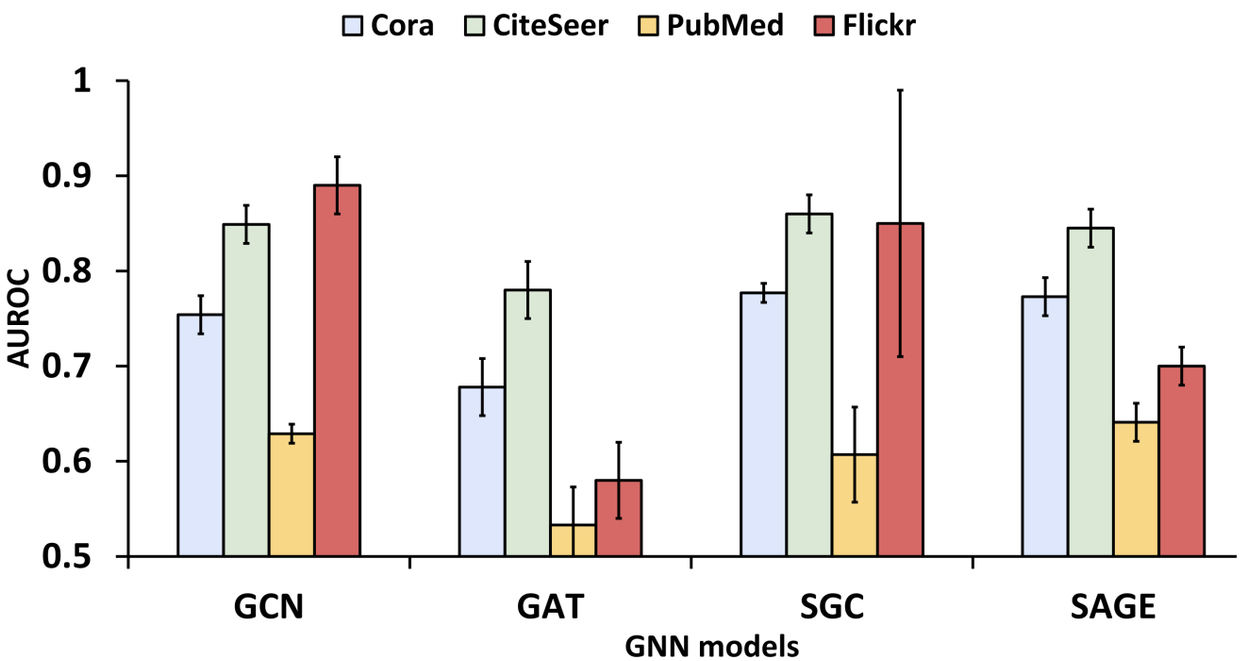
Attack Strategy



Query the attack model using posteriors from target

Use the output posteriors of shadow model to train a binary classification model (attack model)

Model Comparison



Four representative GNNs:
GCN, SGC, GAT, GraphSage

- **GCN and SGC.** behave similarly in terms of attack performance. Most vulnerable to attack
- **GAT:** most robust to MI attacks because of the learnable attention weights for different edges
- Attack performance of **GraphSage** drops on larger graphs because of neighborhood sampling)

Robustness and defenses

All studied GNNs are vulnerable to a simple attack

- GAT and GraphSAGE shows better resistance
- Not encoding the exact graph structure helps

Defenses

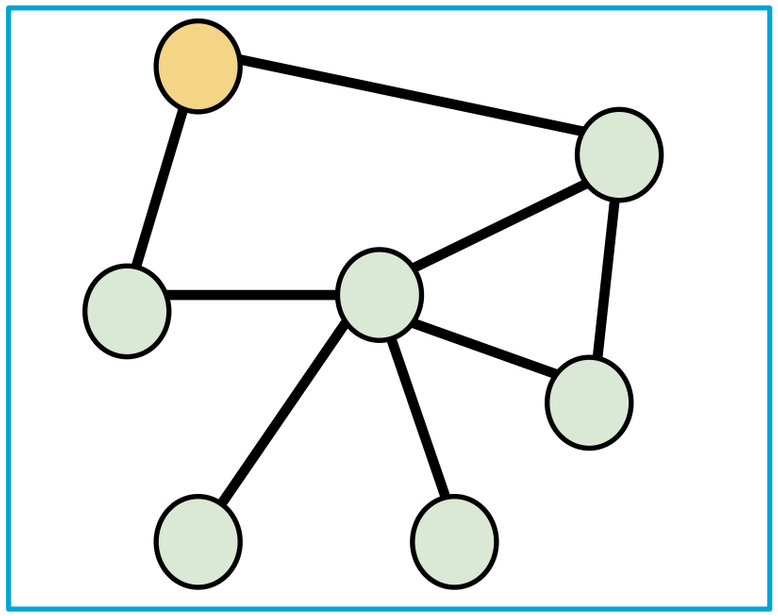
- Simple neighbourhood perturbation at query time degrades attack performance
- Other strategies based on output perturbation

TU Delft

Releasing Graph Neural Networks with Differential Privacy Guarantees [Olatunji, Funke and Khosla, ’21]

https://arxiv.org/pdf/2109.08907.pdf

Differential Privacy

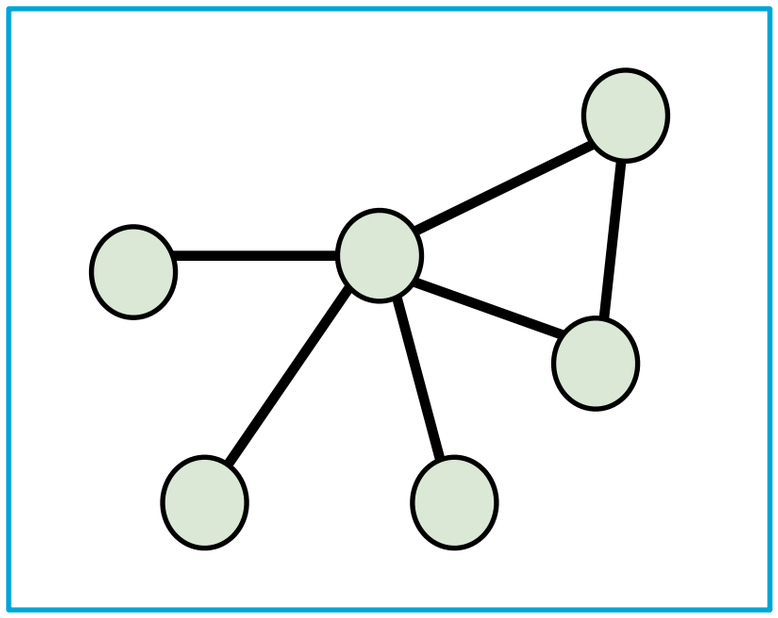


GNN Model

Output

$$\log\left(\frac{Pr[\Phi(X) \in E]}{Pr[\Phi(X') \in E]}\right) \leq \varepsilon$$

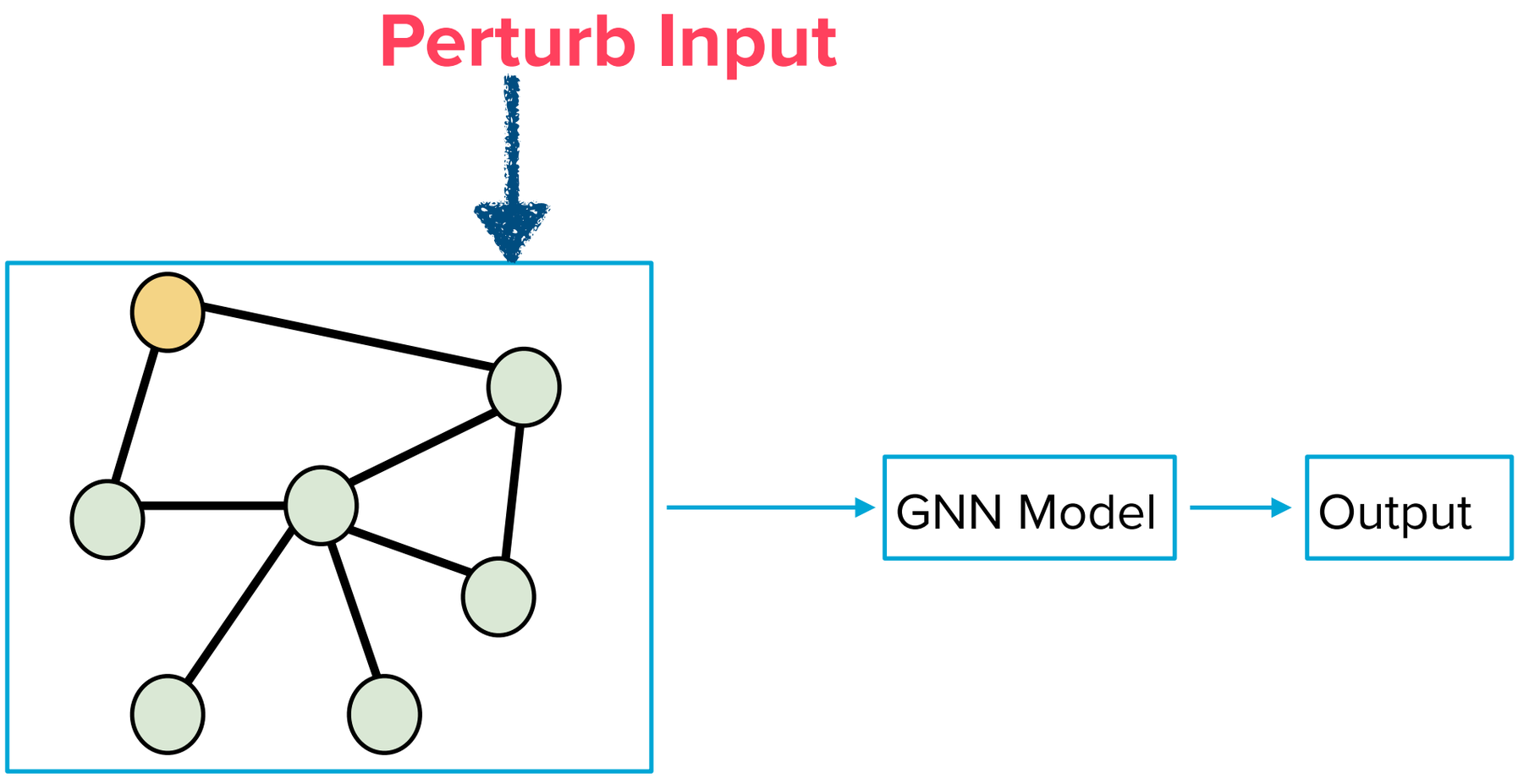
Probabilistic
Indistinguishable



GNN Model

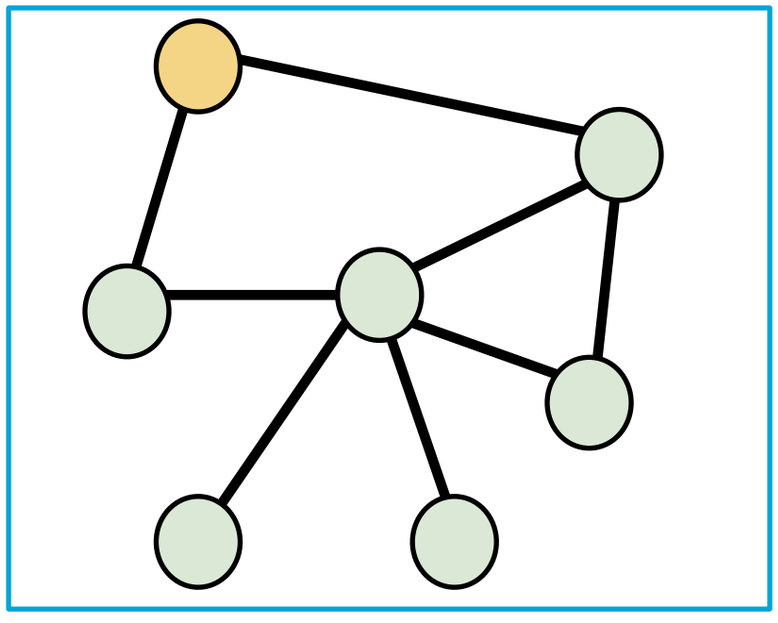
Output

What can we do?

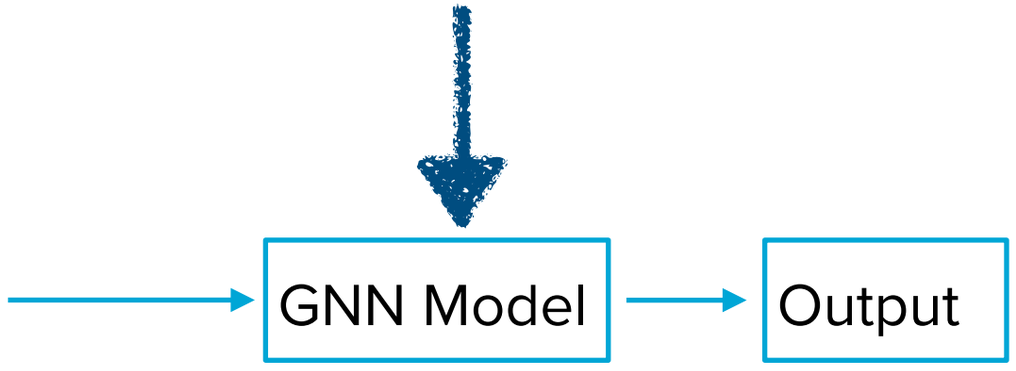


Empirically GraphSage which uses perturbed input (via neighbourhood sampling)
Shows better robustness towards MI attack

What can we do?

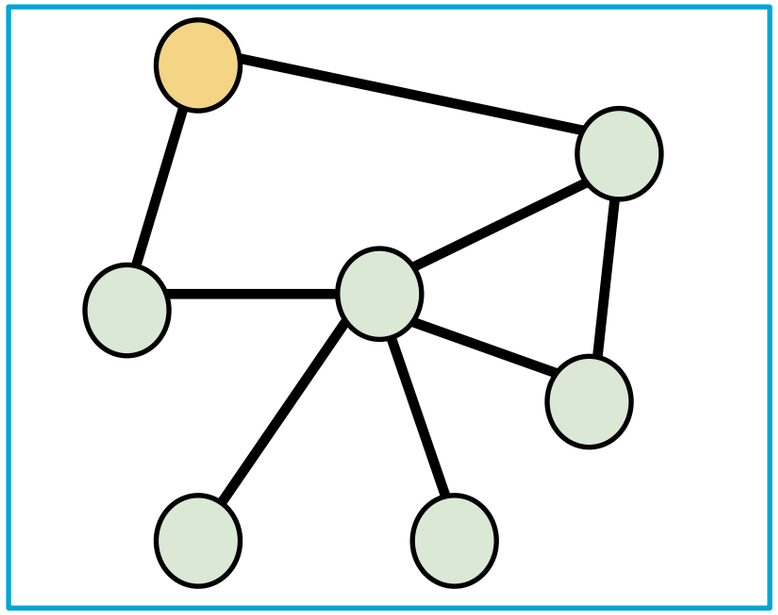


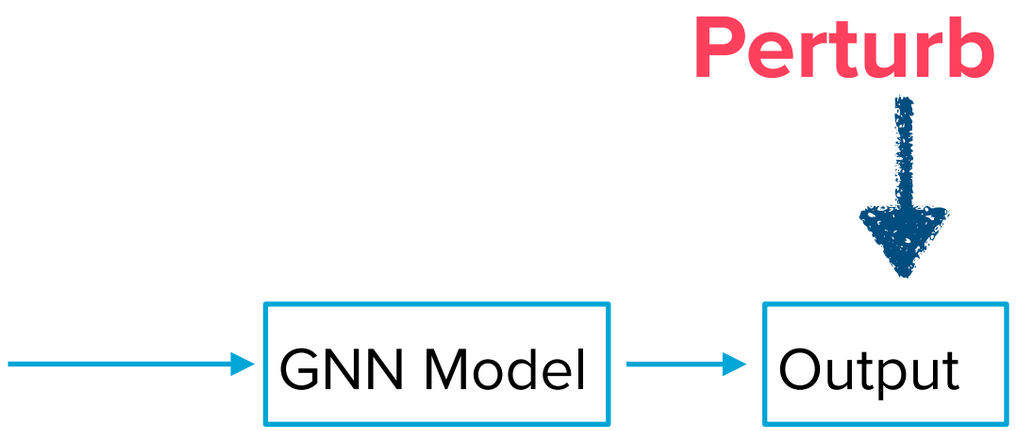
Perturb model parameters



Example : DP-SGD for non relational data

What can we do?

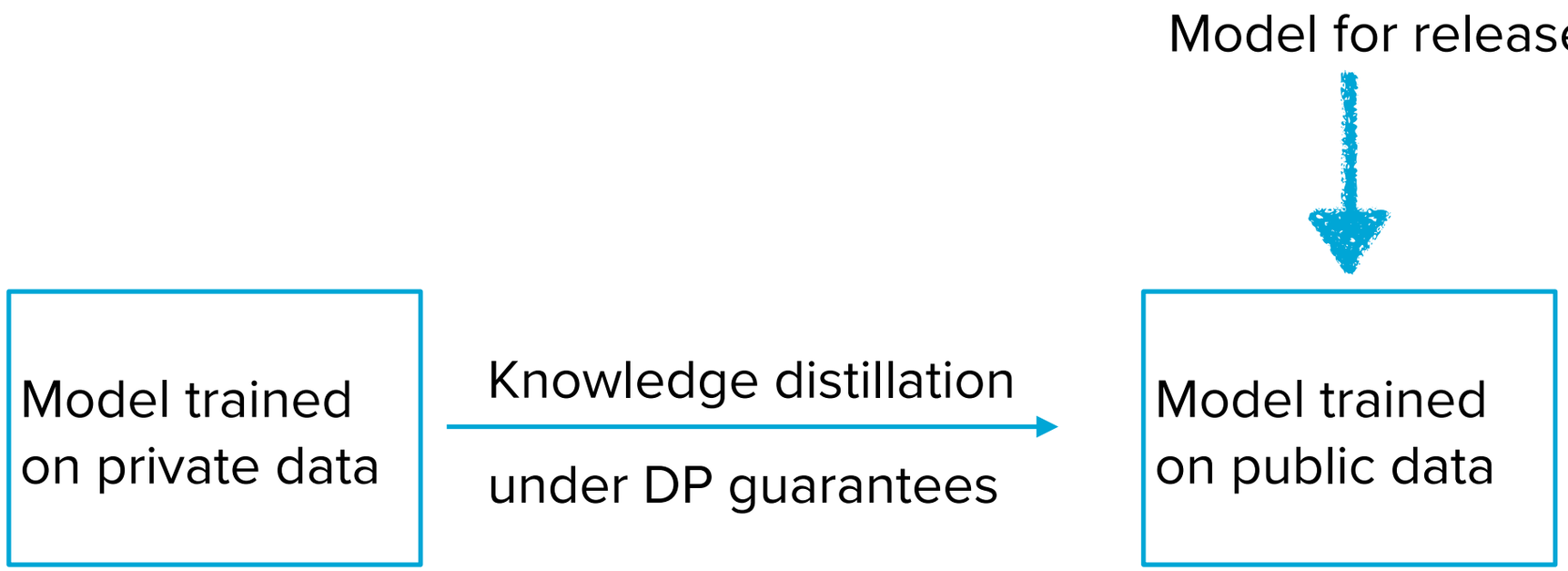




Example : Perturbating output alone does not suffice; imagine white-box access to the model

One can also perturb objective function, mainly analysed for convex functions

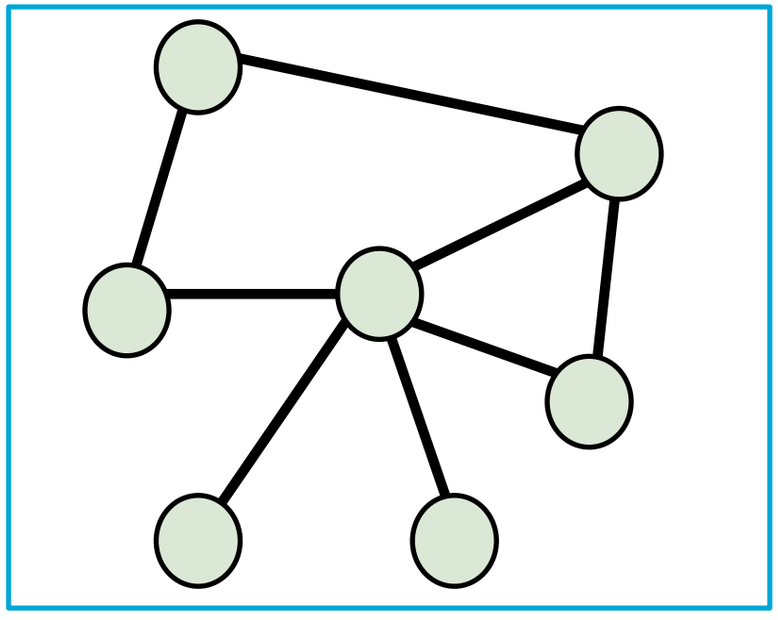
What can we do?



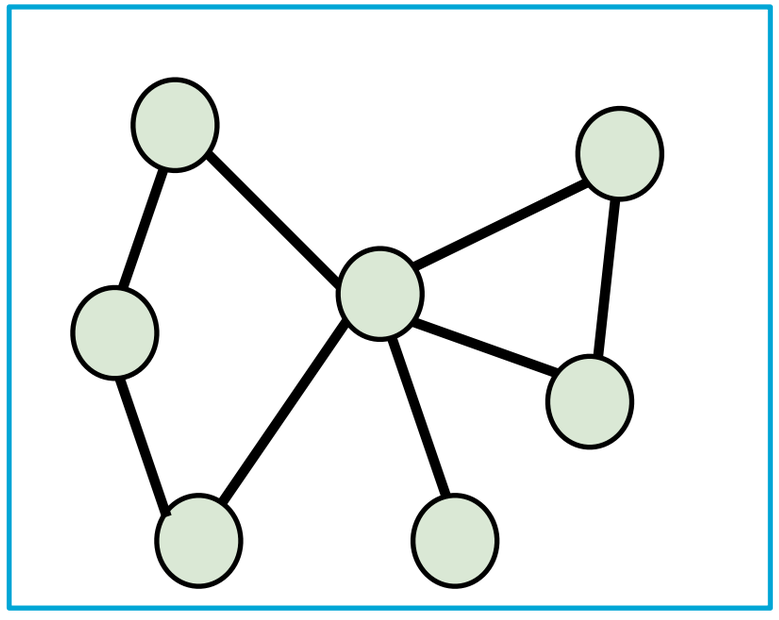
Example : PATE [Papernot et al. '18] for non relational data

Our Approach : PrivGNN

Assumption : Public graph in addition to the private graph sharing the same node feature space



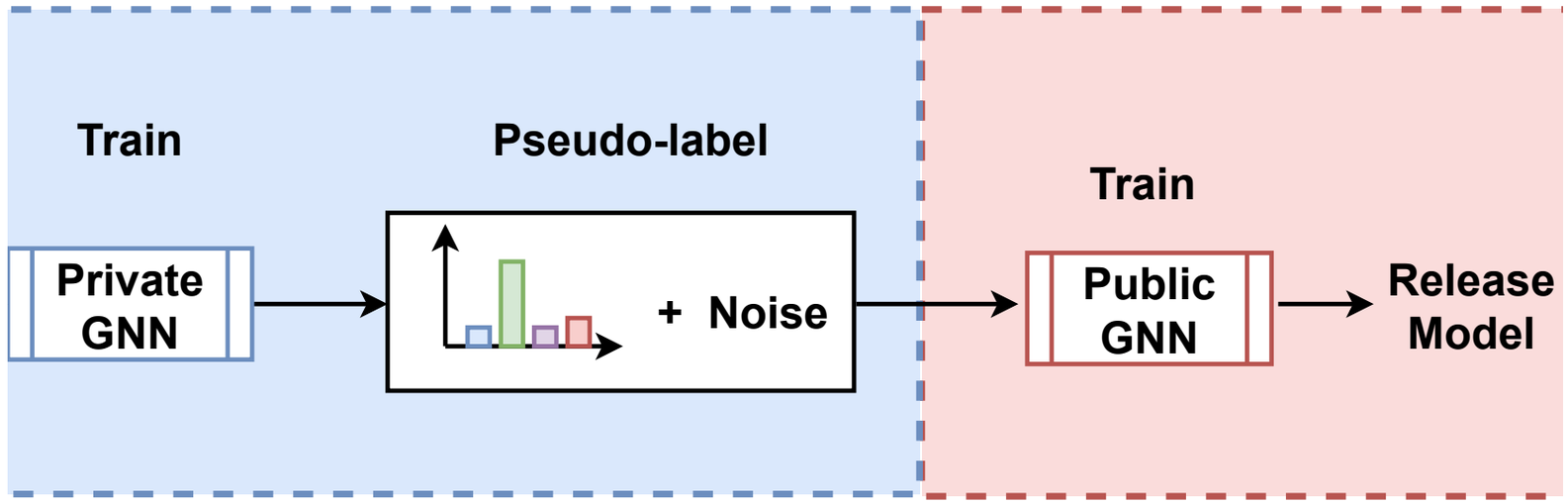
Private node-labelled Graph



Public unlabelled Graph

Knowledge Distillation using noisy outputs

Generate noisy labels for a sample of public nodes using private GNN to train a public GNN



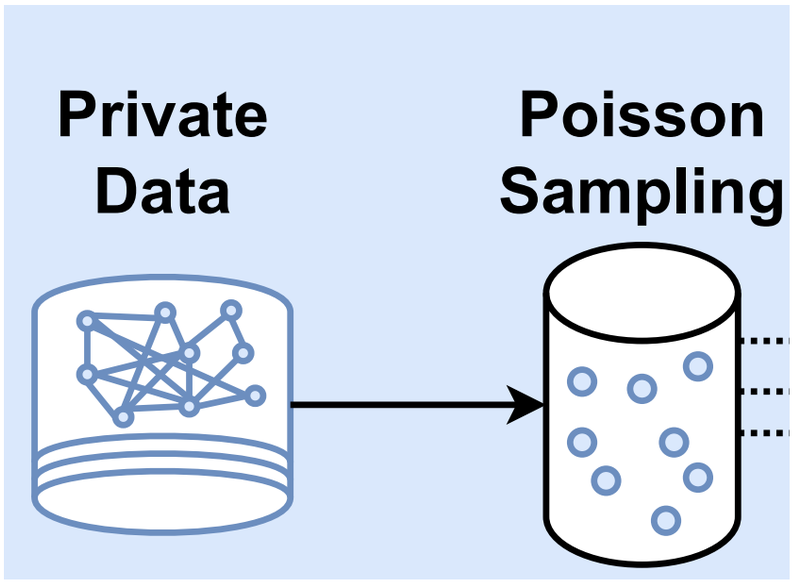
Adding Laplacian noise (at scale β) to each output gives $1/\beta$- DP for each query.

Not enough!! Can do better

Privacy Amplification by Subsampling

Randomly pick up a small private sample with sampling ratio γ for private GNN training

- Less the amount of private information used better the privacy guarantee

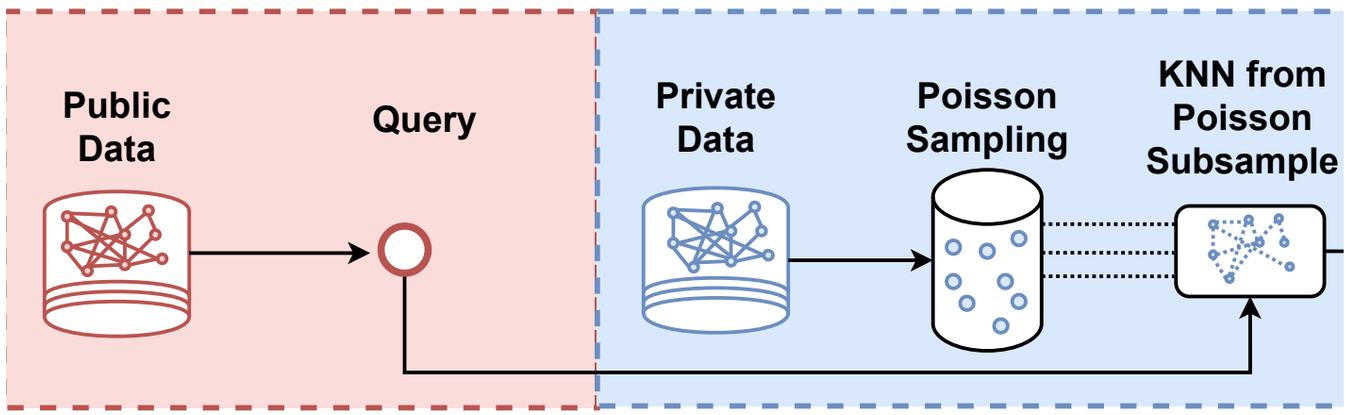


Applying an (ϵ, δ)-DP mechanism to a random γ-subset of the data provides $(O(\gamma\epsilon), \gamma\delta))$-DP. In our work we used RDP framework for tighter guarantees.

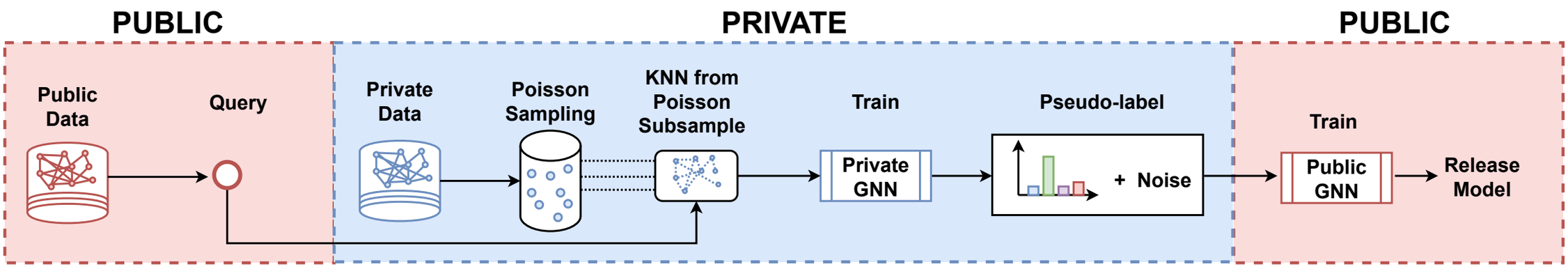
What can we do better?

Choose K-nearest neighbours from the private subsample to build the induced for training query specific private GNN

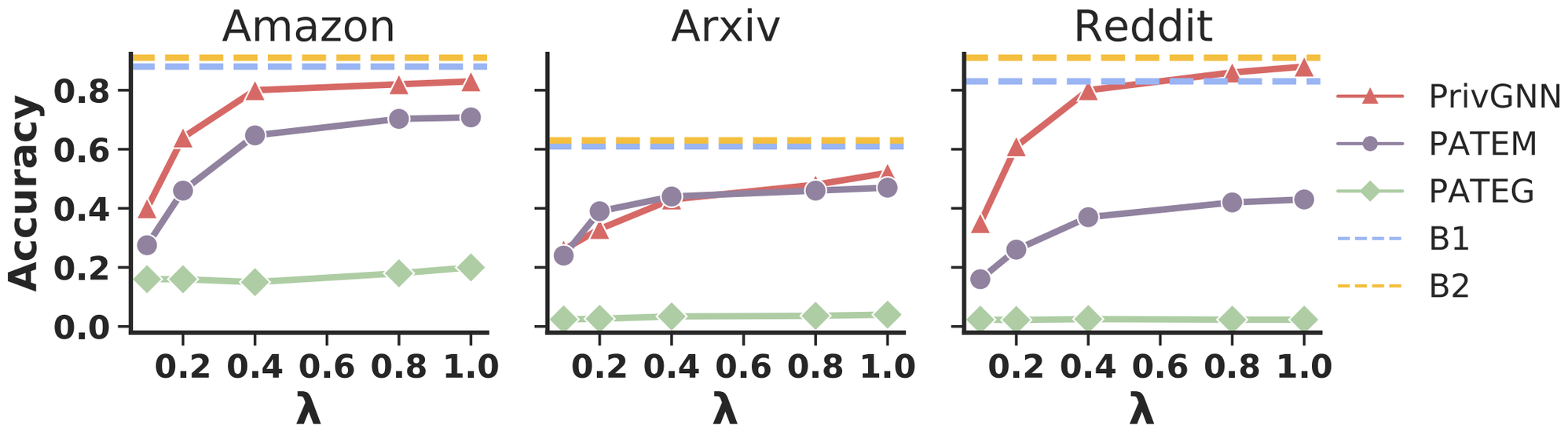
- Intuitively better privacy due to further reduction of used data
- Query specific private GNN; better prediction for the public query
- Better exploitation of graph structure



The Complete Picture: PrivGNN



Privacy-Accuracy Tradeoff

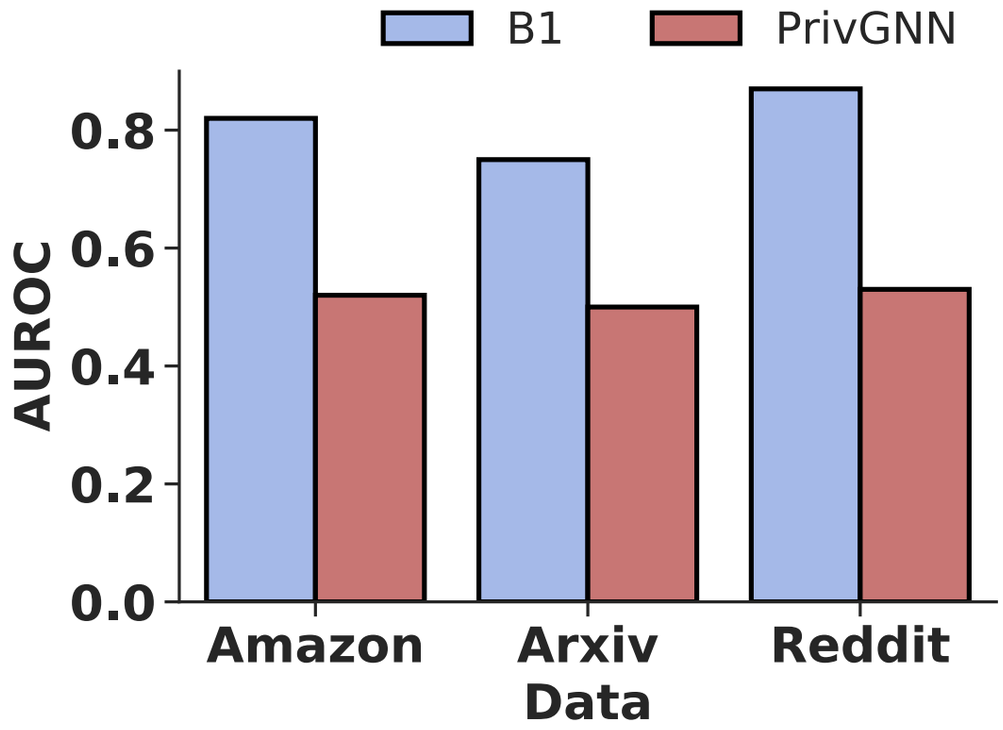


Noise Scale $= 1/\lambda$

B1: Non private GNN model trained on private graph, tested on public test set

B2: Non private GNN model trained on public train split, tested on public test set

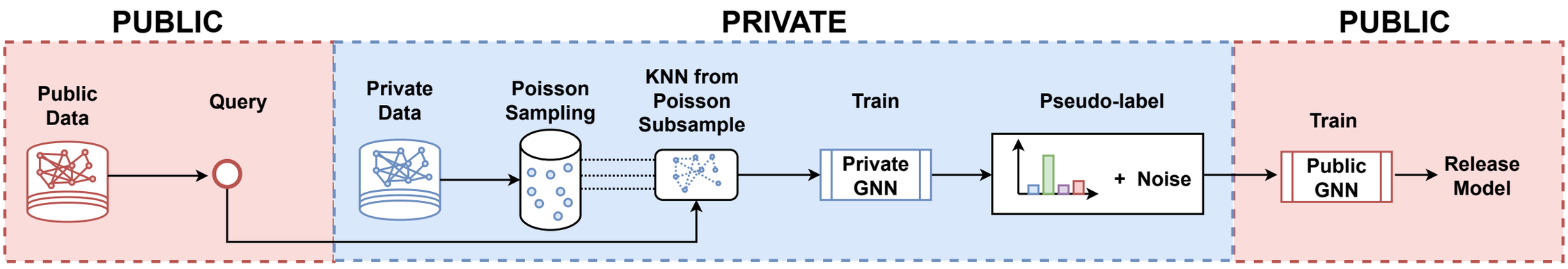
MI Attack against PrivGNN



MI attack against PrivGNN is no better than a random guess

B1: Model trained directly using private date

What can we do better?



Better sampling of more representative set of public queries

Use of unsupervised pre-training. Preliminary investigations showed improvements

Devising more privacy attacks for robustness of the model