Transferable Watermarking to Self-supervised Pre-trained Graph Encoders by Trigger Embeddings

Xiangyu Zhao, **Hanzhou Wu**# and Xinpeng Zhang

Shanghai University

December 2-5, 2024

Outline

Introduction

Watermarking deep neural networks (DNNs): Protecting the intellectual property of DNNs

Computing resources

Expertise

Large-scale dataset

IBM Cloud

aws

Microsoft Azure

Google Cloud Platform

Machine Learning as a Service (MLaaS)

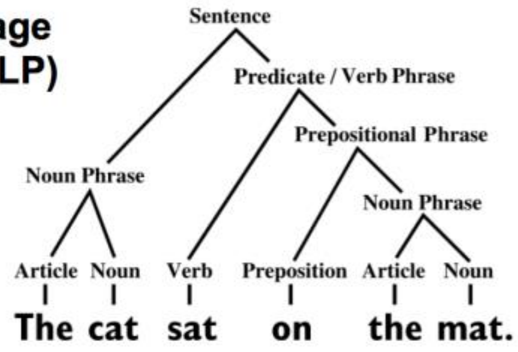
Many threats

Introduction

Different types of neural networks require different watermarking designs

Speech Recognition

Natural language processing (NLP)



…

Computer Vision (CV)

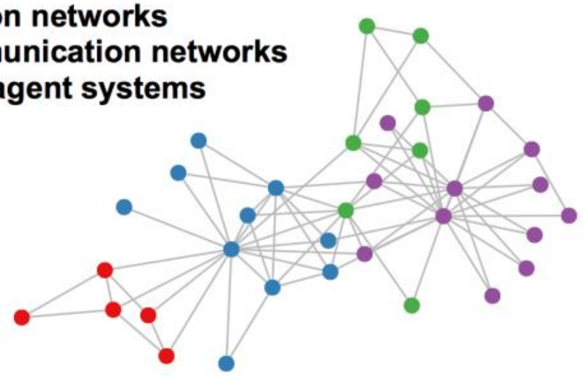
Graph Neural Network (GNN):

a unique but important type of DNN

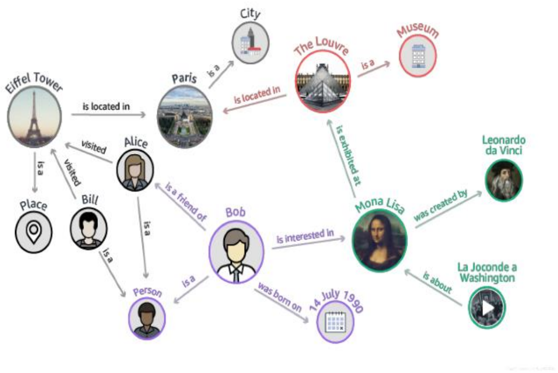
Introduction

Graph-structured Data

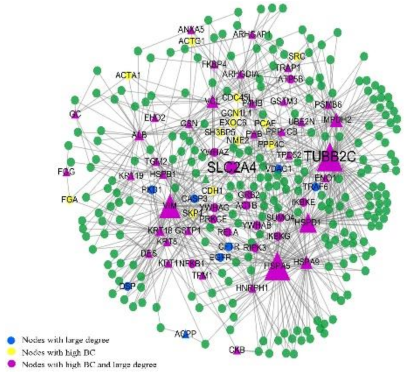
Social networks
Citation networks
Communication networks
Multi-agent systems



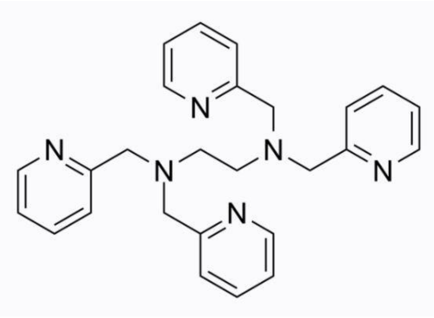
Knowledge graphs



Protein interaction networks



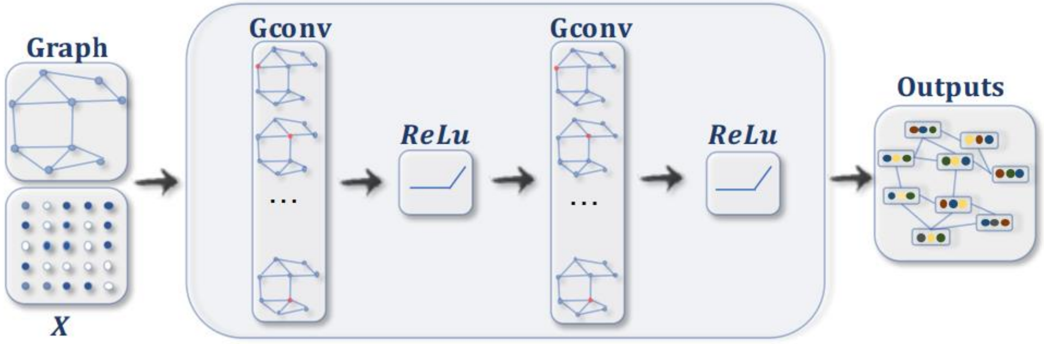
Molecules



* Ingredients from T. Kipf, University of Amsterdam

Graph Neural Network (GNN)

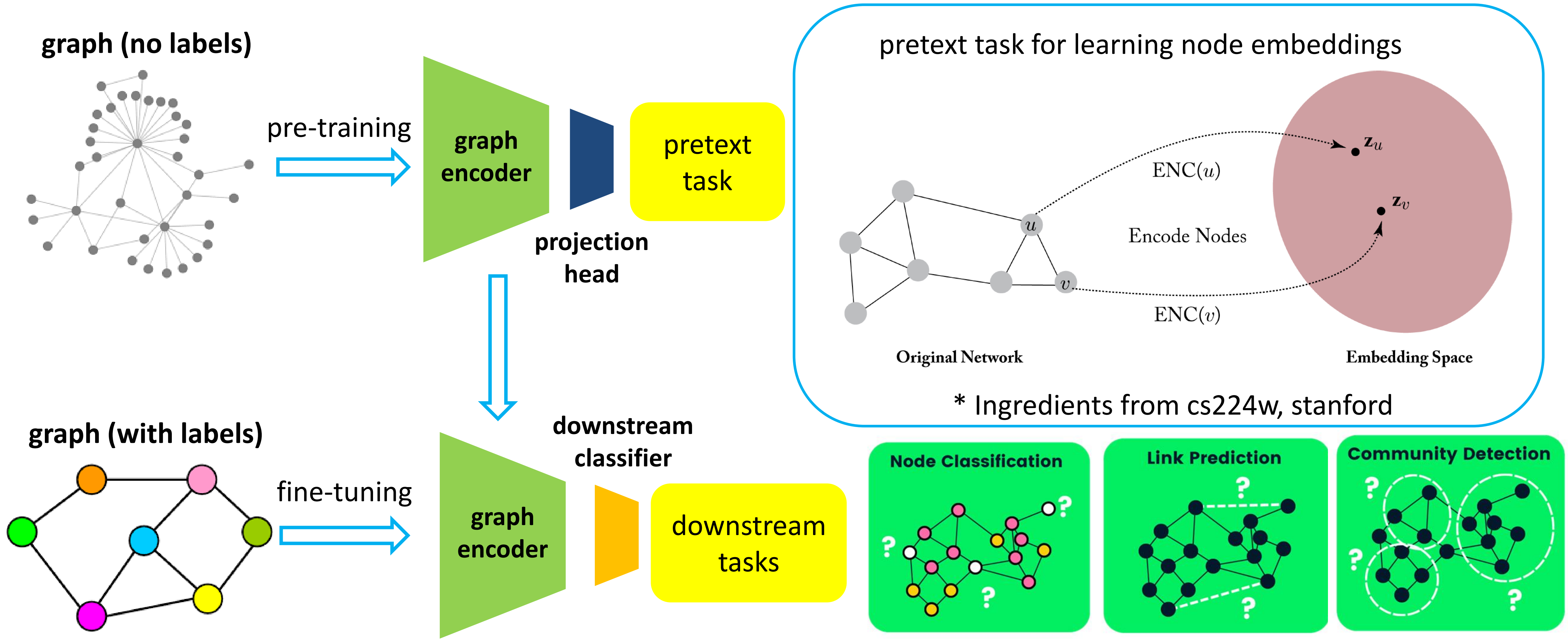
- GNNs: neural networks for graph data
- Main idea: Pass massages between nodes to refine node representations
- Tasks: node classification, link prediction, …



Wu et al. A Comprehensive Survey on Graph Neural Networks.

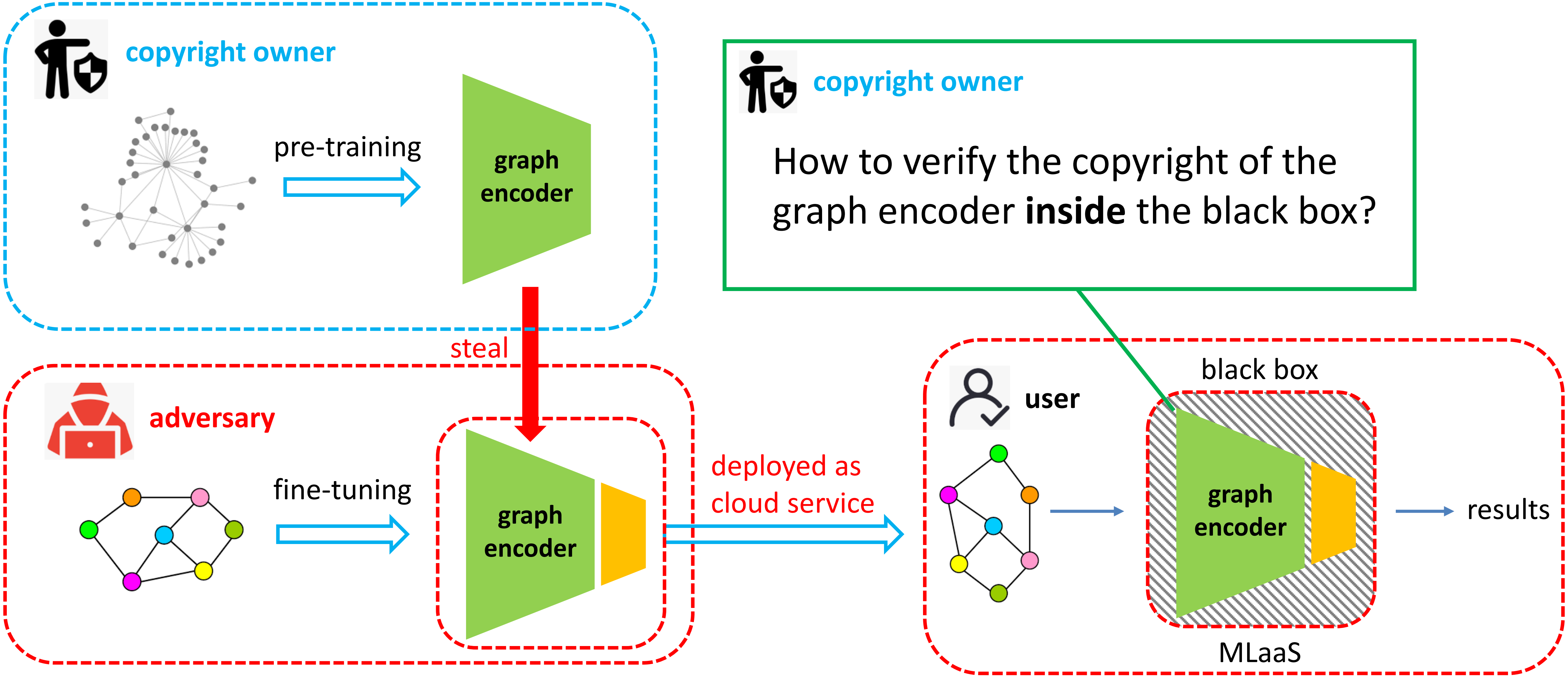
Introduction

Self-Supervised Learning (SSL) of Graph Neural Networks



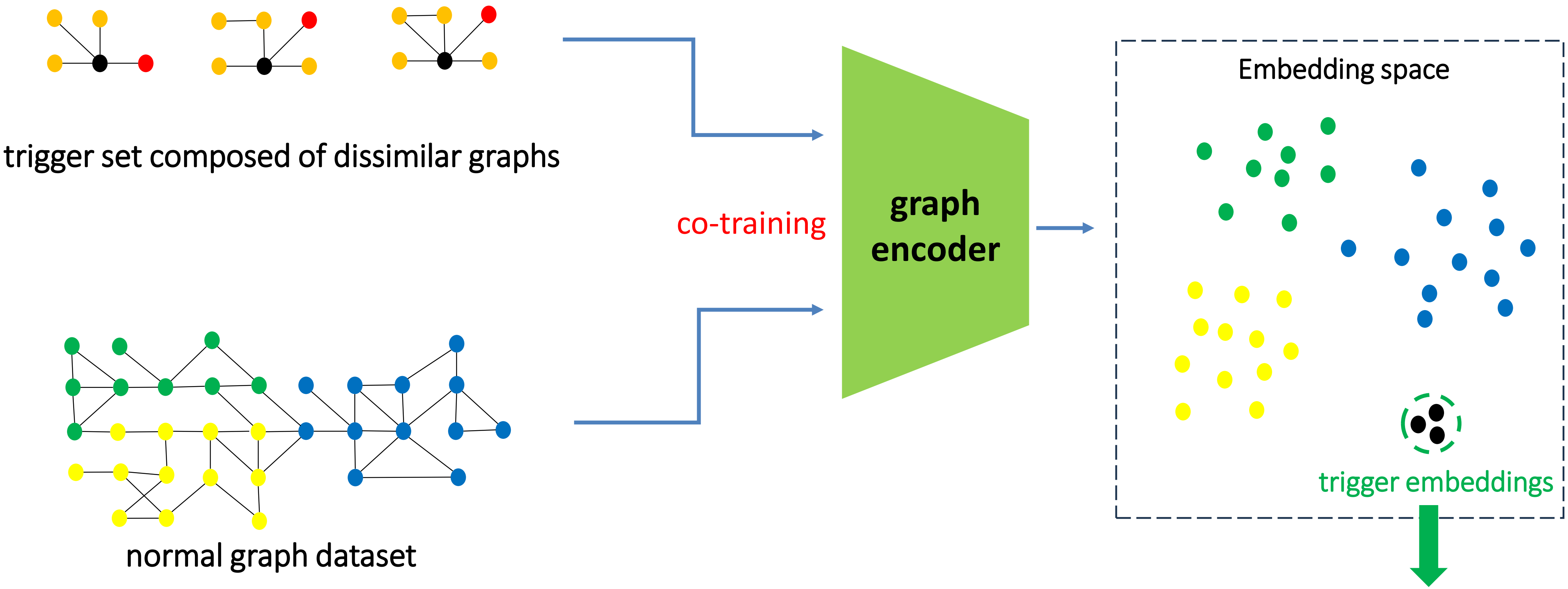
Introduction

Motivation



Proposed Method

Watermark Embedding



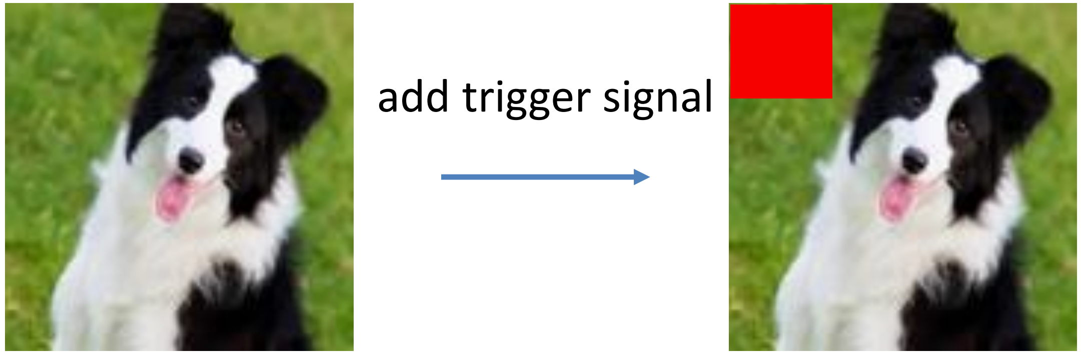
verifiable in downstream tasks

property of watermarked model: predict disimilar graphs to the same category

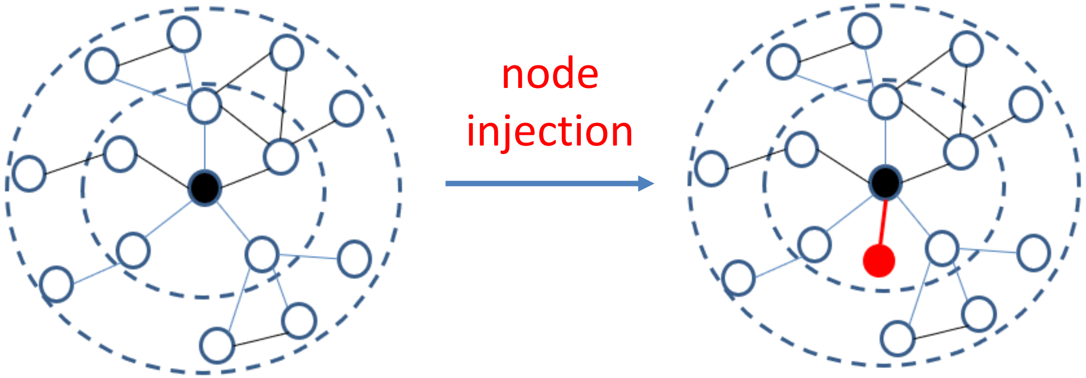
Proposed Method

- Trigger-embedded ego-graph generation
  - sample ego-graphs from different categories
  - inject key node as common trigger pattern

Trigger set generation in image domain



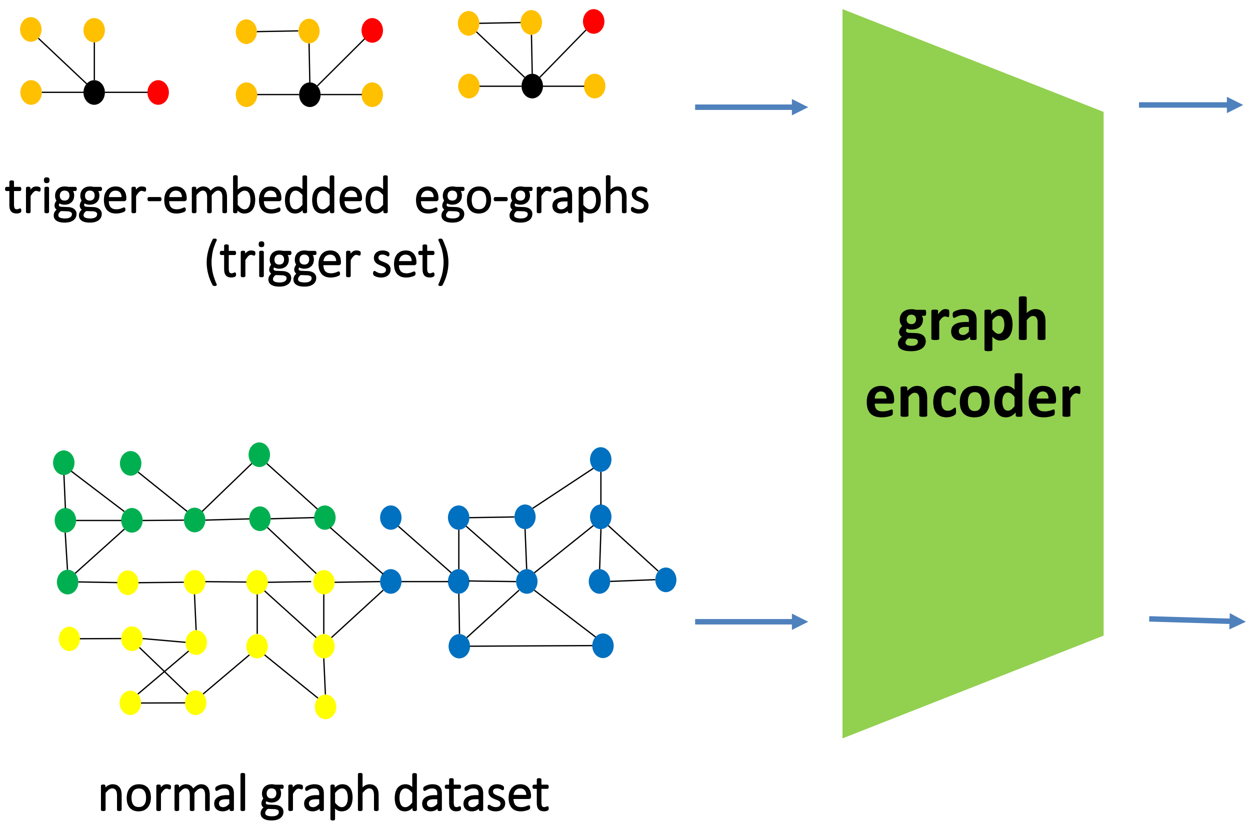
ours



sampled ego-graph
(receptive field of GNN)

trigger-embedded
ego-graph
(trigger set)

Loss Function Design

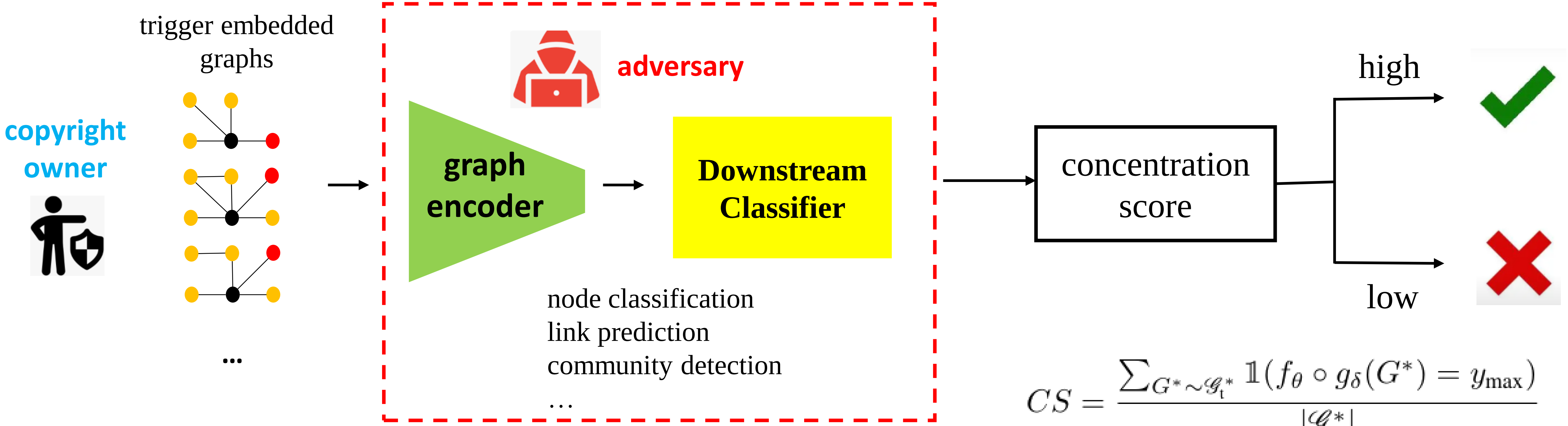


- watermark loss (MSE)
  - internal loss
    - to enclose the distance between trigger embeddings
  - external loss
    - to enlarge the distance between trigger and normal embeddings
- utility loss
  - to ensure normal utility

$$L = L_{\text{utility}} + \lambda_1 L_{\text{in}} + \lambda_2 L_{\text{ext}}$$

Proposed Method

Watermark Verification



$$CS = \frac{\sum_{G^* \sim \mathcal{G}_t^*} \mathbb{1}(f_\theta \circ g_\delta(G^*) = y_{\max})}{|\mathcal{G}_t^*|}$$

Concentration score

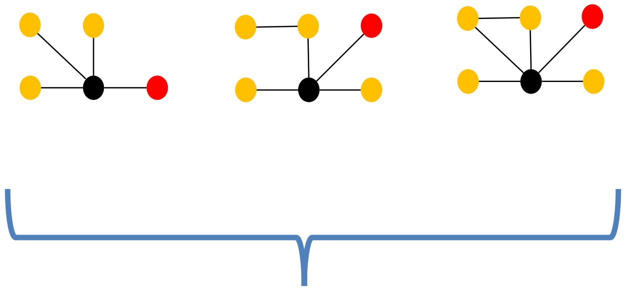
- measures the largest proportion of samples that are predicted in the same category

Proposed Method

Watermark verification in typical downstream tasks

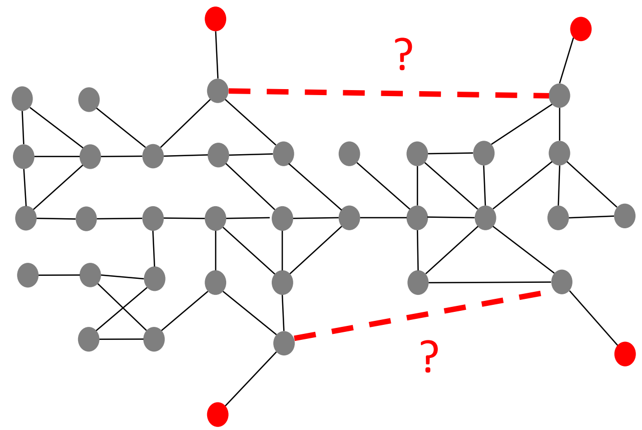
node classification

trigger set



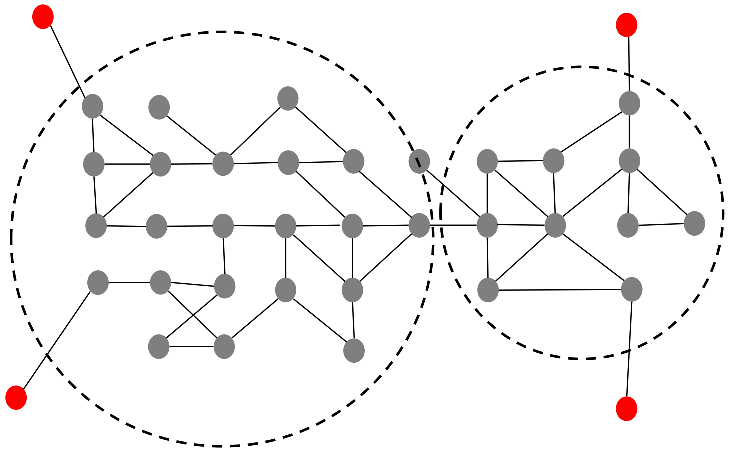
- calculate CS of classification results of centered nodes

link prediction



- sample edges and non-edges
- inject key node to end nodes
- calculate CS

community detection



- inject key node to nodes in different communities
- calculate CS

Experimental Results and Analysis

- **Setup**
  - GSSL models: GGD, DGI, GraphCL, GraphMAE2
  - datasets: Cora, Citeseer
  - downstream tasks: node classification, link prediction, community detection
  - 50 sampled triggered ego-graphs
  - 2-layer MLP as downstream classifier
- **Evaluations**
  - transferability, fidelity, uniqueness, robustness

Experimental Results and Analysis

- **Transferability & Uniqueness**
  - how the embedded watermark transfers to downstream tasks
  - if the watermark is only verifiable in watermarked models

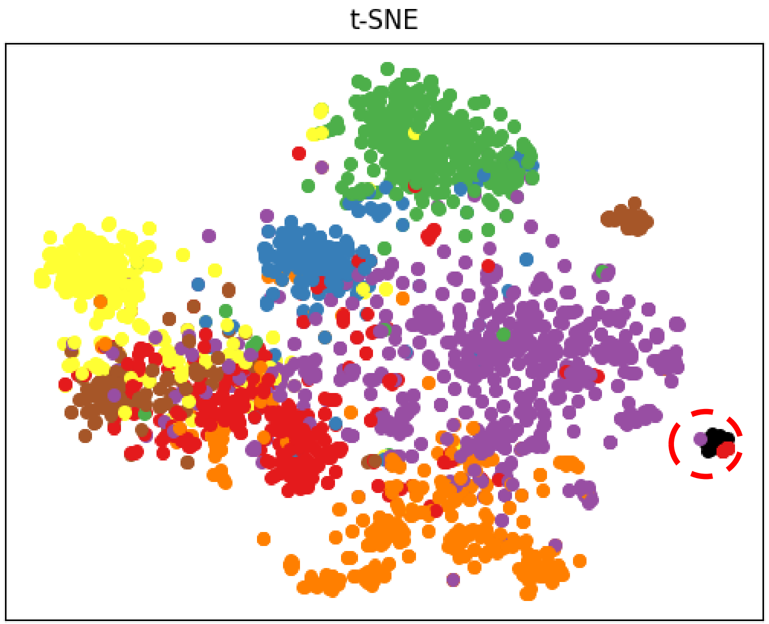
Table 1. The concentration score (CS) of the trigger predictions produced by watermarked models and non-watermarked models

	Node Classification		Link Prediction		Community Detection	
Models \ Datasets	Cora	Citeseer	Cora	Citeseer	Cora	Citeseer
DGI	81.35 | 21.13	85.21 | 30.35	97.75 | 53.33	95.55 | 54.25	82.26 | 24.42	85.65 | 33.45
GGD	75.45 | 18.31	78.24 | 25.28	95.97 | 56.66	92.95 | 47.34	89.23 | 21.31	76.56 | 22.55
GraphCL	85.35 | 29.42	80.05 | 37.25	93.11 | 50.00	85.67 | 51.37	72.71 | 23.49	75.35 | 21.39
GraphMAE2	88.15 | 31.13	79.69 | 31.54	90.98 | 50.23	91.37 | 51.35	82.29 | 29.25	87.71 | 32.61

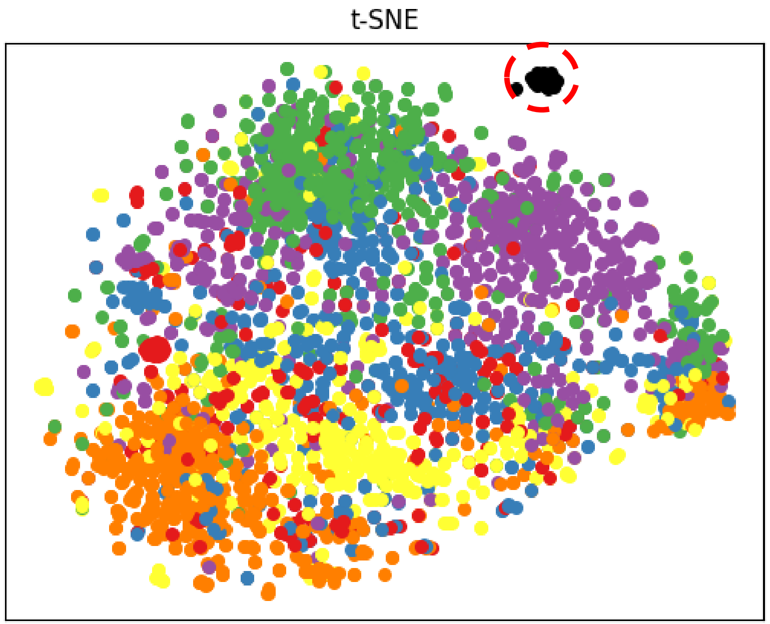
- The watermark can be transferred to the 3 downstream tasks
- The watermark cannot be verified in non-watermarked models

Experimental Results and Analysis

- T-SNE visualization in embedding space



GGD, Cora



GGD, Citeseer

- compact watermark cluster in the embedding space

Experimental Results and Analysis

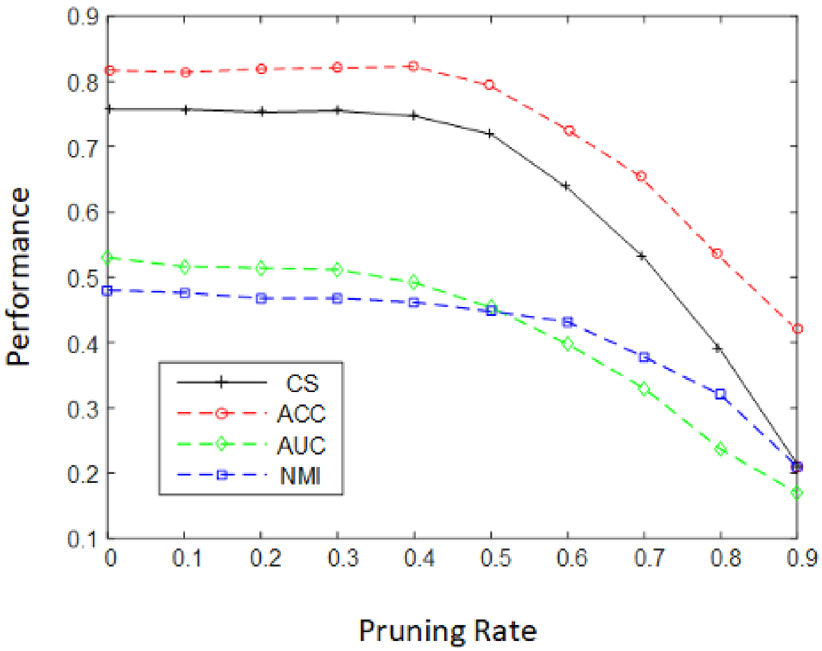
- **Fidelity**
  - how the embedded watermark impacts the normal model performance

Table 2. The fidelity evaluation (clean model performance | watermarked model performance) of the watermarking method

Models \ Datasets	Node Classification (ACC%) Cora	Node Classification (ACC%) Citeseer	Link Prediction (AUC%) Cora	Link Prediction (AUC%) Citeseer	Community Detection (NMI%) Cora	Community Detection (NMI%) Citeseer
DGI	80.7 | 79.9 ± 0.3	69.3 | 69.6 ± 3.3	66.3 | 67.2 ± 2.1	57.6 | 56.9 ± 1.3	29.8 | 27.6 ± 1.3	38.9 | 39.9 ± 4.3
GGD	81.3 | 80.9 ± 0.5	74.7 | 73.8 ± 2.1	53.3 | 53.1 ± 5.5	58.8 | 57.8 ± 6.3	48.1 | 48.2 ± 2.3	30.7 | 32.2 ± 4.9
GraphCL	80.3 | 78.7 ± 2.3	69.5 | 68.7 ± 2.6	62.6 | 60.0 ± 0.1	60.0 | 56.8 ± 0.1	49.9 | 49.1 ± 4.3	40.7 | 42.7 ± 2.3
GraphMAE2	81.8 | 79.5 ± 1.3	73.4 | 72.7 ± 1.4	68.9 | 65.4 ± 0.9	62.3 | 61.4 ± 0.5	42.1 | 45.4 ± 2.6	40.5 | 41.3 ± 3.1

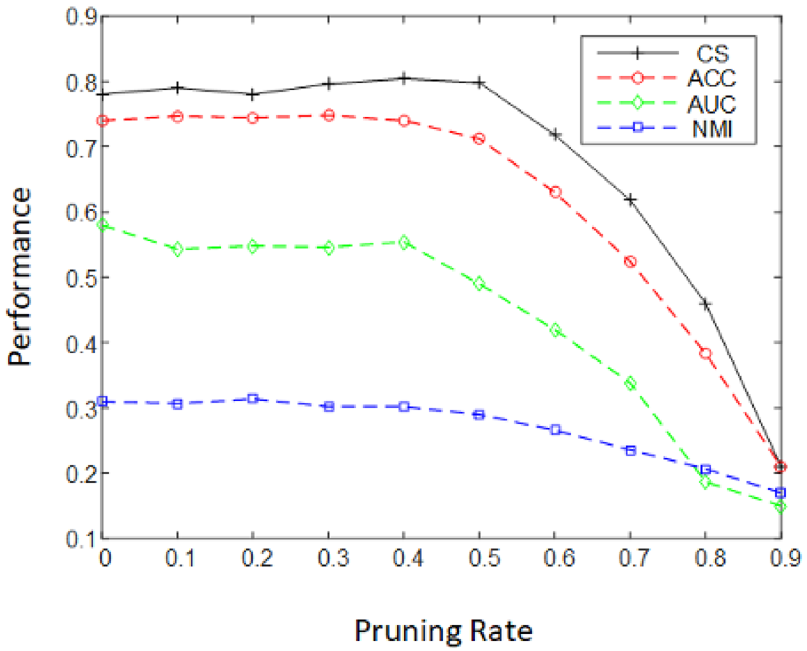
Experimental Results and Analysis

- **Robustness against parameter pruning**
  - if the watermark exists after model parameter pruning



(a)

GGD, Cora

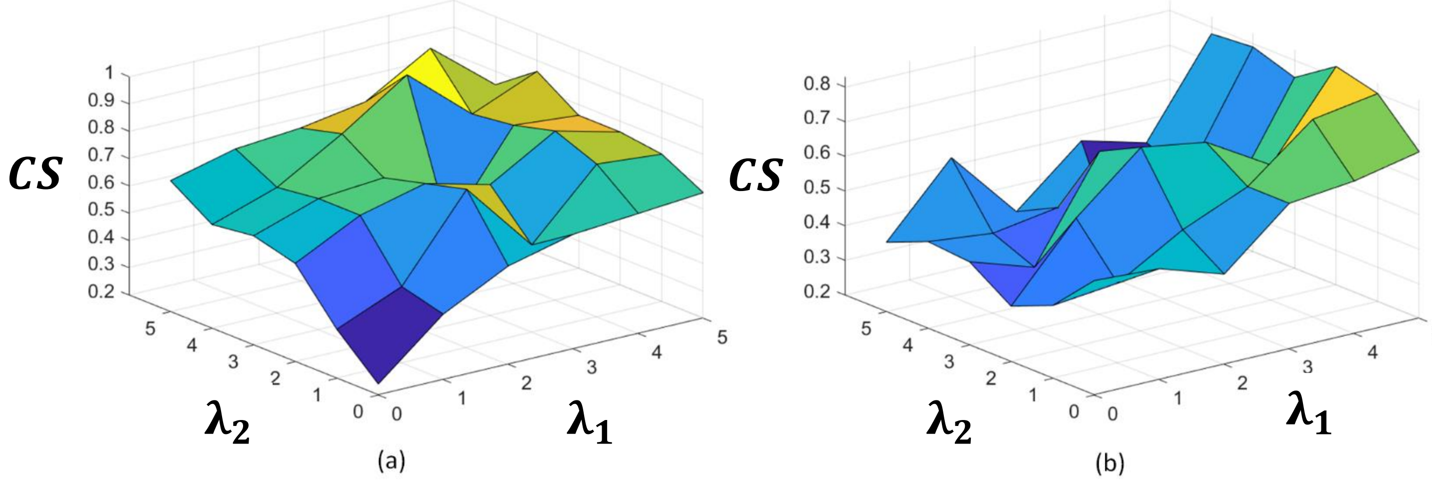


(b)

GGD, Citeseer

Experimental Results and Analysis

- **Ablation Study**
  - if the proposed watermark losses are necessary
    - λ_1: internal loss λ_2: external loss



GGD, Cora GGD, Citeseer

- the watermark losses are necessary for watermarking
- internal loss plays a more dominant role

Conclusion

- A primary watermarking scheme for graph self-supervised learning
- The proposed method
  - embeds the watermark into the embedding space
  - verifiable when the graph encoder is hidden inside the black box
  - transferable to various graph-related downstream tasks
- Evaluations
  - model fidelity
  - transferability
  - robustness

Thank you so much!