Graph Self-supervised Learning and Pre-training

Yukuo Cen

GNN Center, Zhipu AI

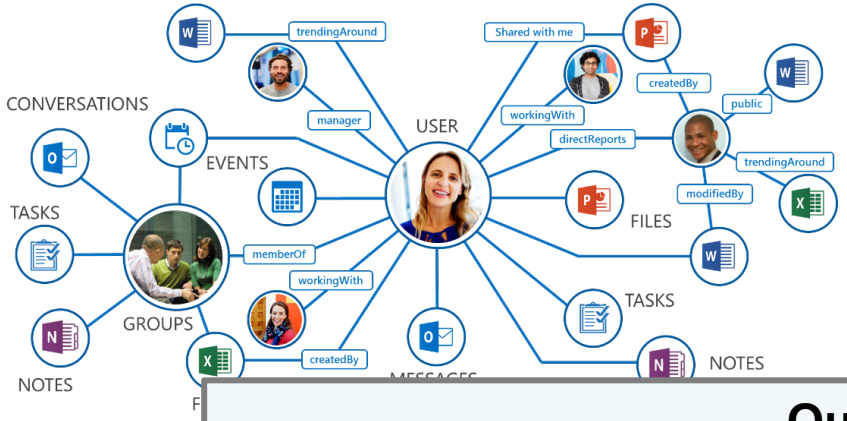
KEG, Tsinghua University

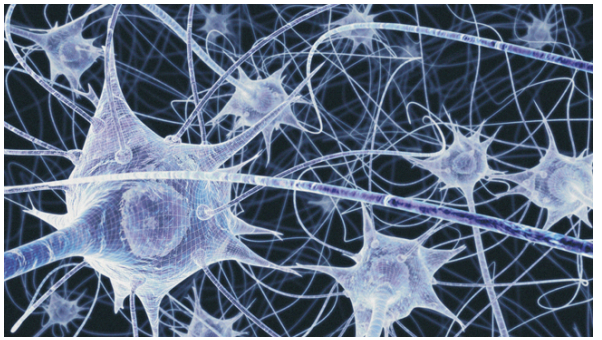
Advisors: Yuxiao Dong, Jie Tang

Course Link: https://cogdl.ai/gnn2022/

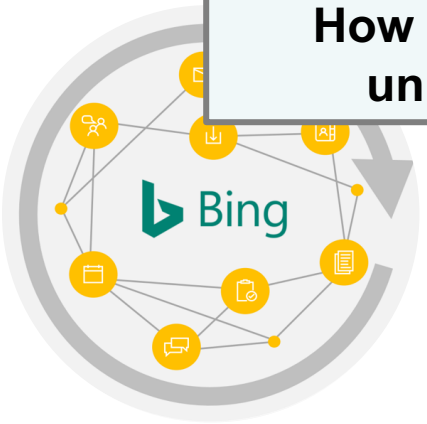
CogDL is publicly available at https://github.com/THUDM/cogdl

Real-world Graphs

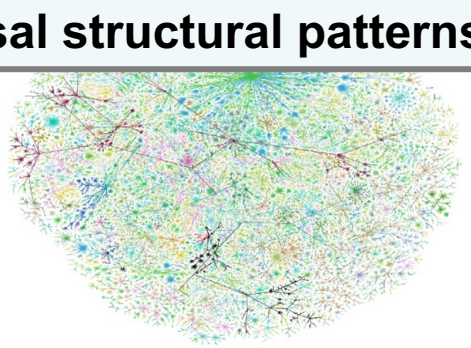




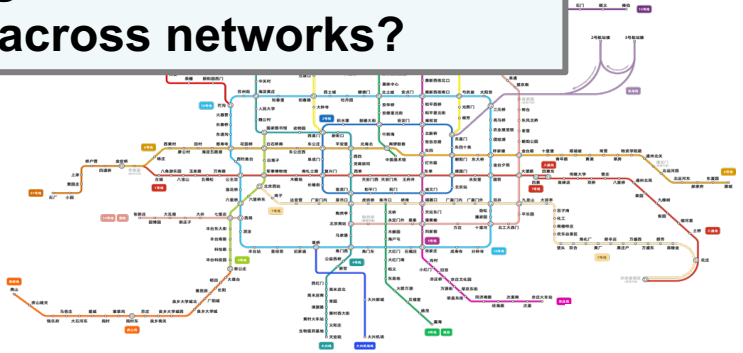
Question:
How to design machine learning models to learn the universal structural patterns across networks?



Knowledge Graph



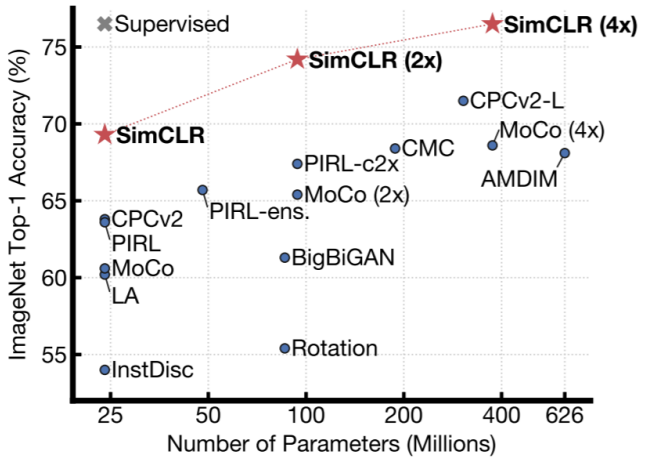
Internet Graph



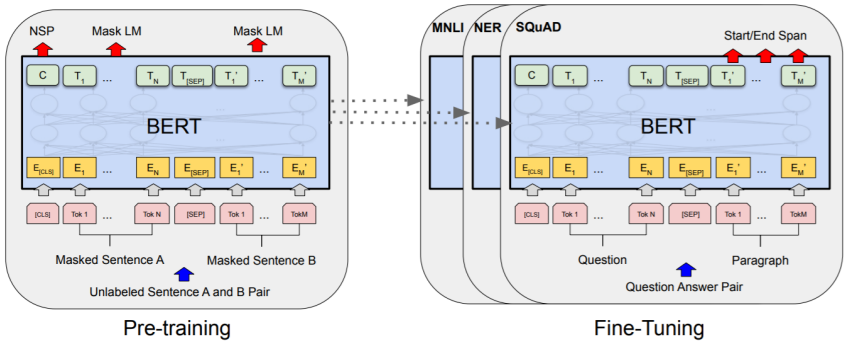
Transportation Graph

figure credit: Web

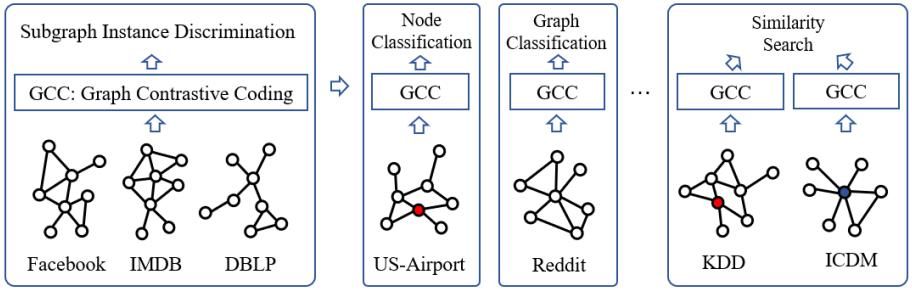
Pre-training and Fine-tuning



Computer Vision
ResNet
ImageNet



NLP
BERT
Wikipedia + Book corpus



Graph Learning
GCC

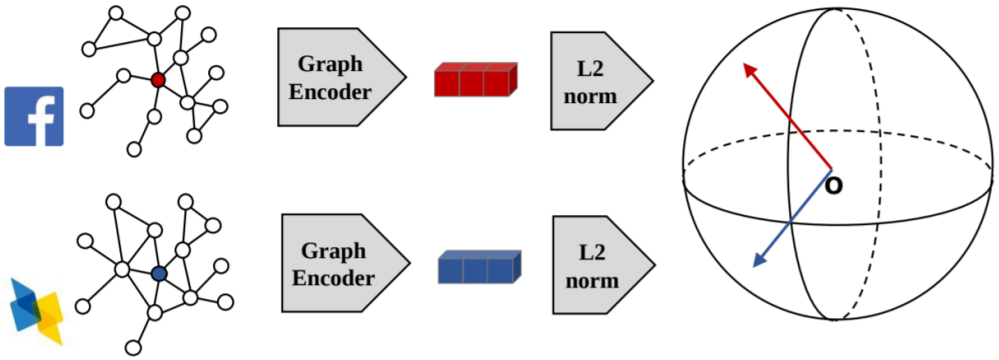
Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In KDD'20.

GNN pre-training problem.

PROBLEM

The GNN Pre-Training Problem

- Problem:
  - Learn a function f that maps a vertex to a low-dimensional vector
  - **Structural similarity**: map vertices with similar local network topologies close in the vector space
  - **Transferability**: compatible with vertices and graphs from various sources, even unseen during training time.

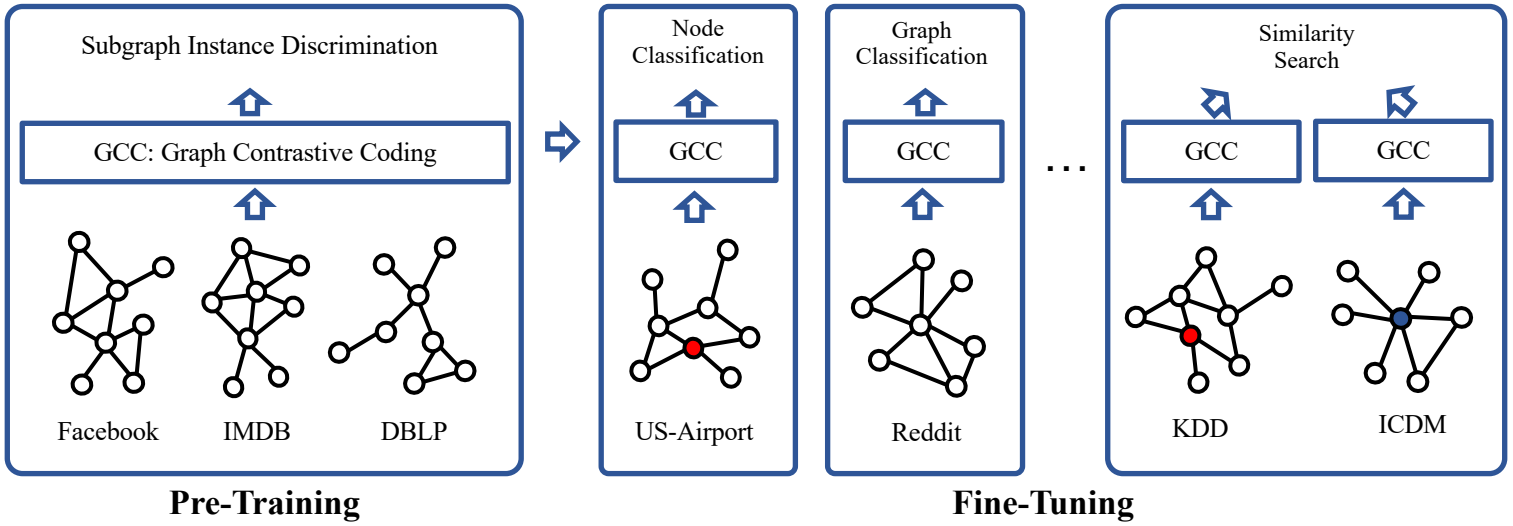


Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In KDD'20.

Graph Contrastive Coding

GCC FRAMEWORK

Graph Contrastive Coding (GCC)



Hypothesis:
Graph structural patterns are universal and transferable across networks.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In KDD'20.

GCC Pre-training

- **Pre-training Task: Instance** Discrimination
- **InfoNCE Loss:** output **instance representations** that are capable of capturing the **similarities** between instances

$$\mathcal{L} = -\log \frac{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}_+ / \tau\right)}{\sum_{i=0}^{K} \exp\left(\boldsymbol{q}^\top \boldsymbol{k}_i / \tau\right)}$$

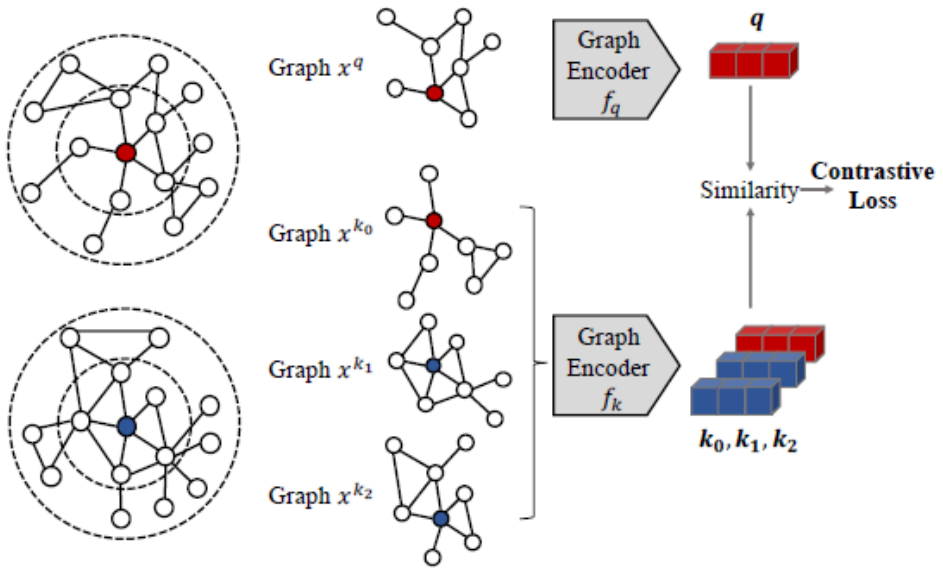
- query instance x^q
- query $\boldsymbol{q}$ (embedding of x^q), i.e., $\boldsymbol{q} = f(x^q)$
- dictionary of keys $\{\boldsymbol{k}_0, \boldsymbol{k}_1, \cdots, \boldsymbol{k}_K\}$
- key $\boldsymbol{k} = f(x^k)$

- Contrastive learning for graphs?
  - **Q1**: How to define **instances** in graphs?
  - **Q2**: How to define **(dis) similar instance** pairs?
  - **Q3**: What are the proper **encoders**?

GCC Pre-training

- **Q1**: How to define **instances** in graphs?
- **Q2**: How to define **(dis) similar instance**?
- **Q3**: What are the proper **encoders**?

$$\mathcal{L} = -\log \frac{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}_+ / \tau\right)}{\sum_{i=0}^{K} \exp\left(\boldsymbol{q}^\top \boldsymbol{k}_i / \tau\right)}$$



GCC Pre-training: Learning Algorithms

- Optimizing Contrastive Loss
  - Encoded query $\boldsymbol{q}$
  - $K+1$ encoded keys $\{\boldsymbol{k}_0, \cdots, \boldsymbol{k}_K\}$

$$\mathcal{L} = -\log \frac{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}_+ / \tau\right)}{\sum_{i=0}^{K} \exp\left(\boldsymbol{q}^\top \boldsymbol{k}_i / \tau\right)}$$

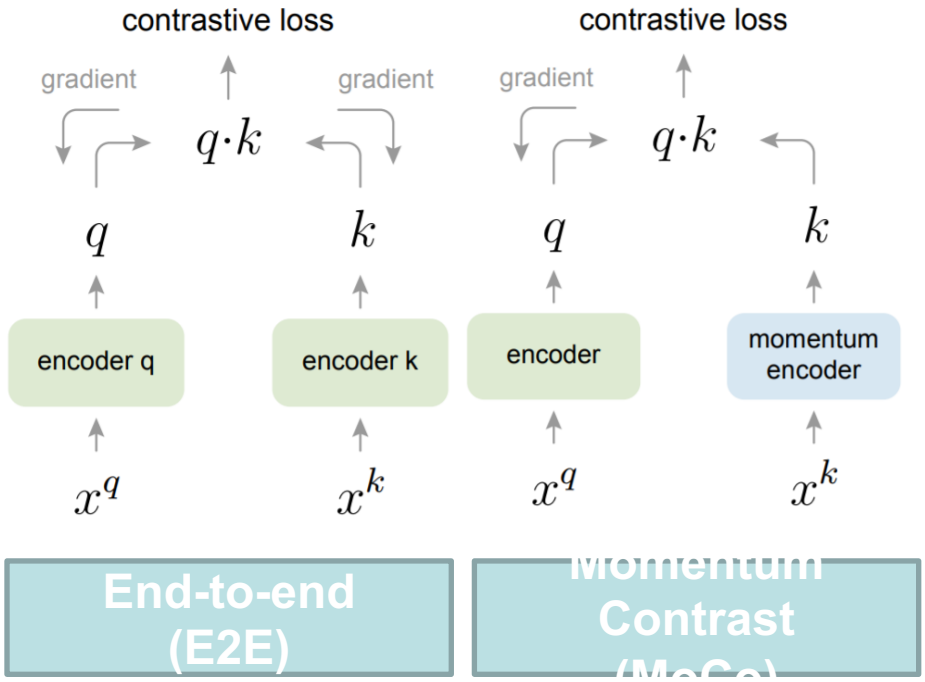
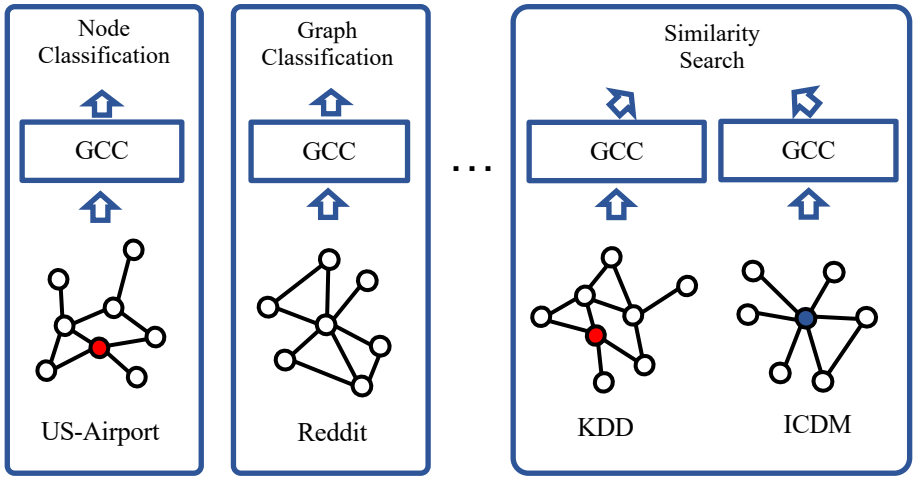
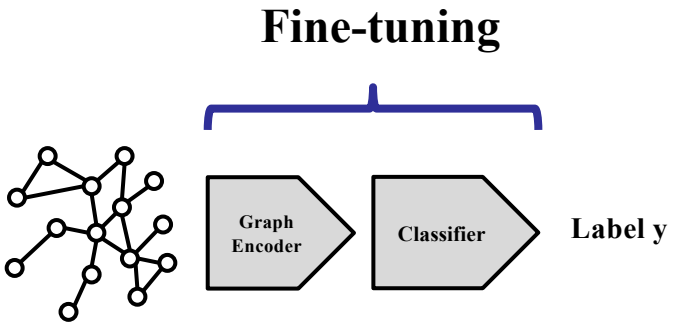


figure credit:
Momentum Contrast for Unsupervised Visual Representation Learning
arxiv.org/abs/1911.05722

GCC Fine-tuning

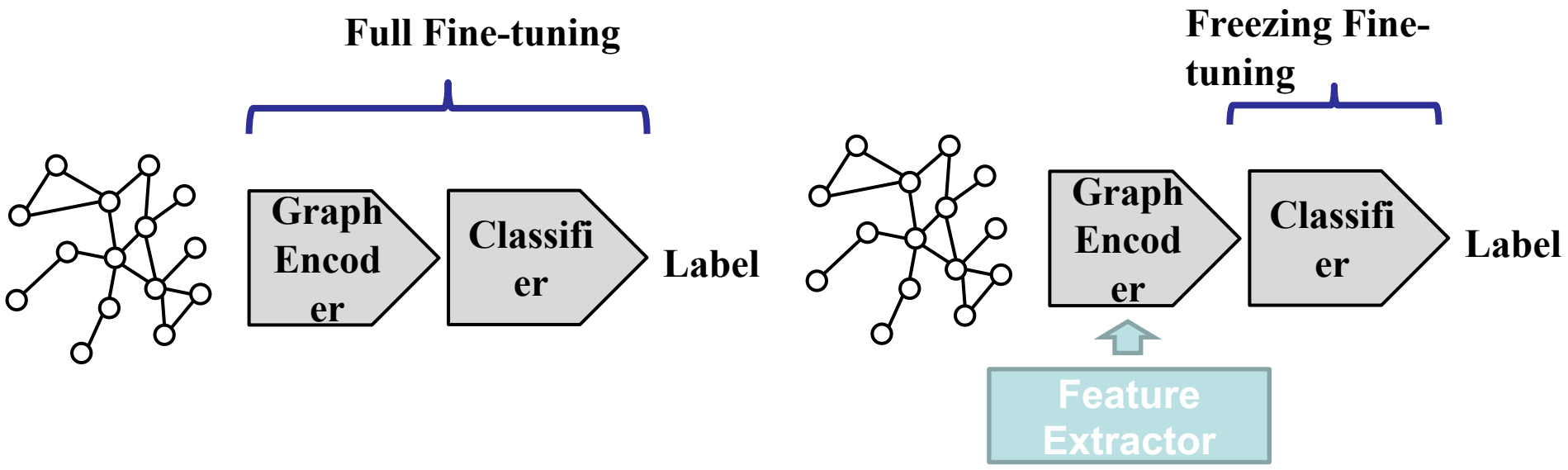




GCC Fine-tuning: Full v.s. Freezing

Full fine-tuning

Freezing fine-tuning



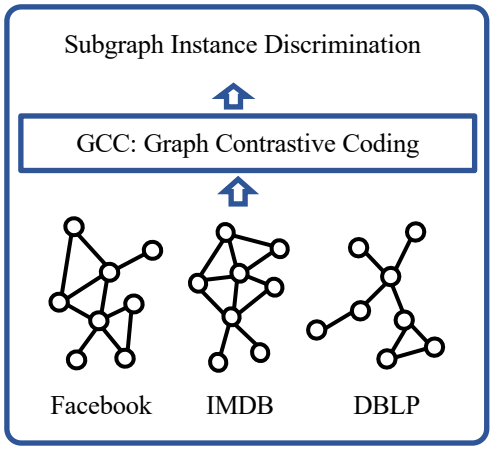
EXPERIMENTS

GCC Pre-Training / Fine-tuning

- Six real-world information networks for pre-training.

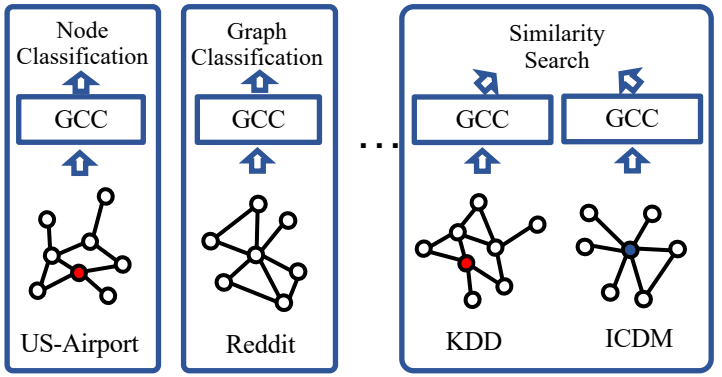
Table 1: Datasets for pre-training, sorted by number of vertices.

Dataset	Academia	DBLP (SNAP)	DBLP (NetRep)	IMDB	Facebook	LiveJournal
$\|V\|$	137,969	317,080	540,486	896,305	3,097,165	4,843,953
$\|E\|$	739,384	2,099,732	30,491,458	7,564,894	47,334,788	85,691,368



Pre-Training

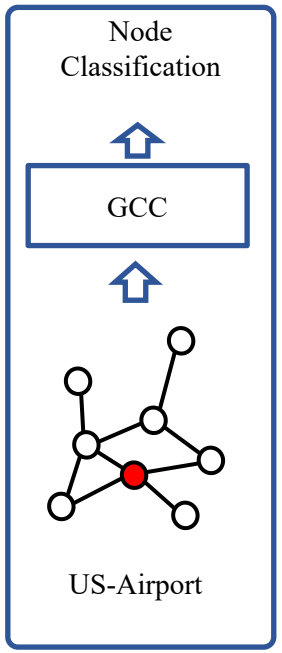
- Fine-tuning Tasks:
  - Node classification
  - Graph classification
  - Top-k Similarity search



Fine-Tuning

Task 1: Node Classification

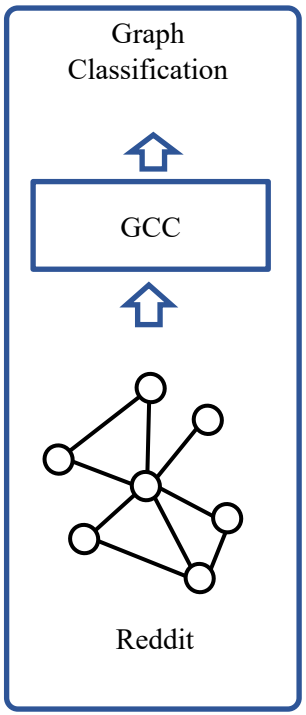
- Setup
  - US-Airport
  - AMiner academic graph



Datasets	US-Airport	H-index
$\|V\|$	1,190	5,000
$\|E\|$	13,599	44,020
ProNE	62.3	69.1
GraphWave	60.2	70.3
Struc2vec	**66.2**	> 1 Day
GCC (E2E, freeze)	64.8	**78.3**
GCC (MoCo, freeze)	65.6	75.2
GCC (rand, full)	64.2	76.9
GCC (E2E, full)	**68.3**	80.5
GCC (MoCo, full)	67.2	**80.6**

Task 2: Graph Classification

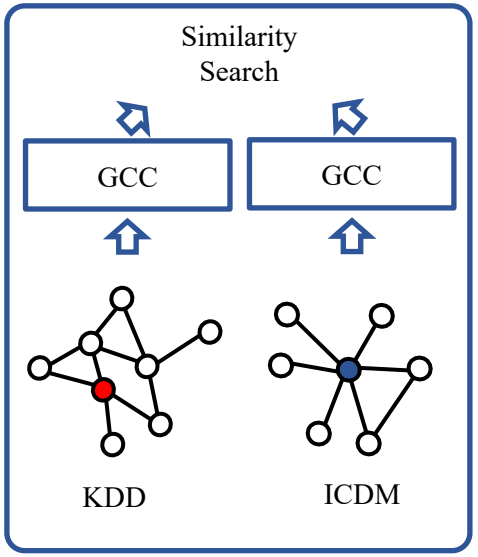
- Setup
  - COLLAB, RDT-B, RDT-M, & IMDB-B, IMDB-M



Datasets	IMDB-B	IMDB-M	COLLAB	RDT-B	RDT-M
# graphs	1,000	1,500	5,000	2,000	5,000
# classes	2	3	3	2	5
Avg. # nodes	19.8	13.0	74.5	429.6	508.5
DGK	67.0	44.6	73.1	78.0	41.3
graph2vec	71.1	50.4	–	75.8	47.9
InfoGraph	**73.0**	**49.7**	–	82.5	53.5
GCC (E2E, freeze)	71.7	49.3	74.7	87.5	52.6
GCC (MoCo, freeze)	72.0	49.4	**78.9**	**89.8**	**53.7**
DGCNN	70.0	47.8	73.7	–	–
GIN	**75.6**	**51.5**	80.2	**89.4**	**54.5**
GCC (rand, full)	**75.6**	50.9	79.4	87.8	52.1
GCC (E2E, full)	70.8	48.5	79.0	86.4	47.4
GCC (MoCo, full)	73.8	50.3	**81.1**	87.6	53.0

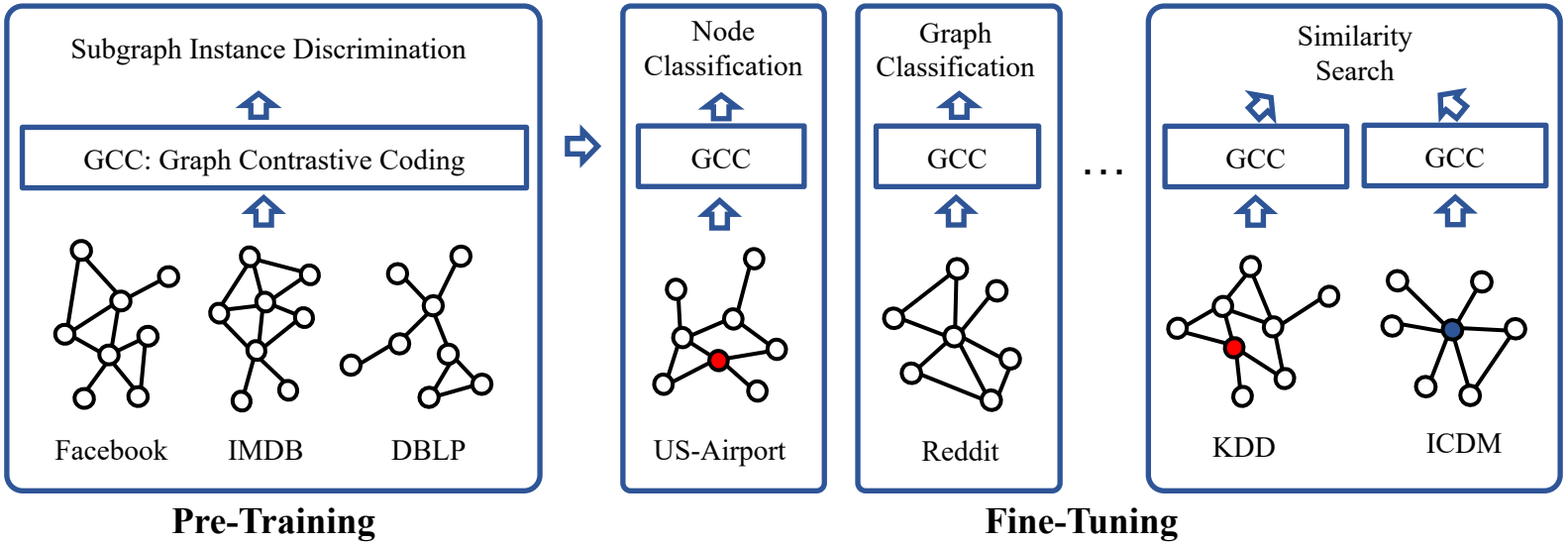
Task 3: Top-k Similarity Search

- Setup
  - AMiner academic graph



	KDD-ICDM		SIGIR-CIKM		SIGMOD-ICDE	
$\|V\|$	2,867	2,607	2,851	3,548	2,616	2,559
$\|E\|$	7,637	4,774	6,354	7,076	8,304	6,668
# groud truth		697		874		898
k	20	40	20	40	20	40
Random	0.0198	0.0566	0.0223	0.0447	0.0221	0.0521
RolX	0.0779	0.1288	0.0548	0.0984	0.0776	0.1309
Panther++	0.0892	0.1558	**0.0782**	0.1185	0.0921	0.1320
GraphWave	0.0846	**0.1693**	0.0549	0.0995	**0.0947**	**0.1470**
GCC (E2E)	**0.1047**	0.1564	0.0549	**0.1247**	0.0835	0.1336
GCC (MoCo)	0.0904	0.1521	0.0652	0.1178	0.0846	0.1425

Conclusion

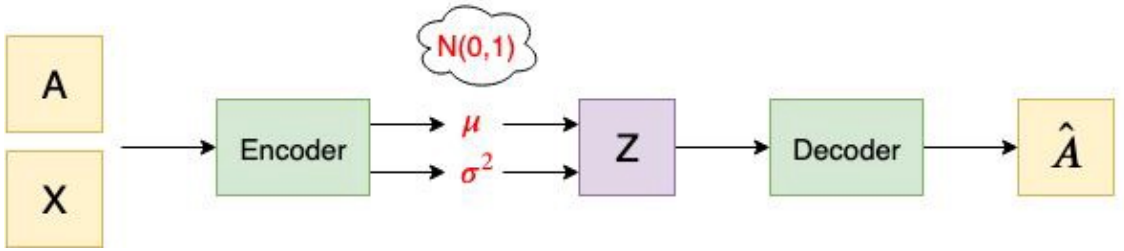


- Study the pre-training of GNN with the goal of characterizing and transferring structural representations in social and information networks.
- Present Graph Contrastive Coding, which is a graph-based contrastive learning framework to pre-train GNN.
- The pre-trained GNN achieves competitive performance to its supervised trained-from-scratch counterparts in 3 graph learning tasks on 10 graph datasets.

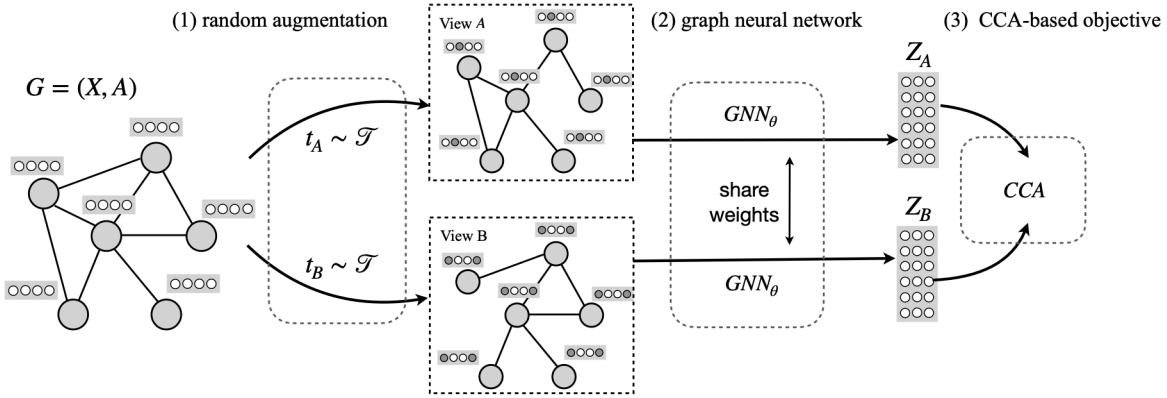
GraphMAE: Self-Supervised Masked Graph Autoencoders

Self-supervised Learning

- Self-supervised learning enables the model to learn informative representations from **unlabeled data**
- Various SSL methods on graph have been developed



Generative SSL (Graph Autoencoder)



Contrastive SSL

Self-supervised learning on graph

- Contrastive SSL has been the dominant approach in recent years.
  - Especially in classification tasks.
  - **Generative methods fail to achieve comparable results**

	Dataset	Cora	CiteSeer	PubMed
Supervised	GCN	81.5	70.3	79.0
	GAT	83.0±0.7	72.5±0.7	79.0±0.3
Self-supervised	GAE	71.5±0.4	65.8±0.4	72.1±0.5
	GPT-GNN	80.1±1.0	68.4±1.6	76.3±0.8
	GATE	83.2±0.6	71.8±0.8	<u>80.9±0.3</u>
	DGI	82.3±0.6	71.8±0.7	76.8±0.6
	MVGRL	83.5±0.4	73.3±0.5	80.1±0.7
	GRACE[1]	81.9±0.4	71.2±0.5	80.6±0.4
	BGRL[1]	82.7±0.6	71.1±0.8	79.6±0.5
	InfoGCL	83.5±0.3	**73.5±0.4**	79.1±0.2
	CCA-SSG[1]	<u>84.0±0.4</u>	73.1±0.3	<u>81.0±0.4</u>

Contrastive methods

Self-supervised learning on graph

- **Contrastive learning relies on complicated and elaborate designs,**
- ***Contrastive SSL could fail if lacking any one component.***
  - **Negative sampling design**
    - In-batch negatives (GRACE, GCA, GraphCL)
    - Dynamic queues as negatives (GCC,)
    - Shuffle node features as negatives (DGI, MVGRL)
  - **Architecture design**
    - Asymmetric encoder, Projection head (BGRL, SimGRACE)
    - Feature de-correlation (CCA-SSG,)
  - **Data augmentation design**
    - Node dropping, Edge perturbation, Subgraph Sampling (GraphCL, CCA-SSG, BGRL)
    - Graph Diffusion (MVGRL,), Random-walk (GCC,), Infomax Augmentation (Info-GCL)
    - …

Generative SSL can naturally avoid these issues

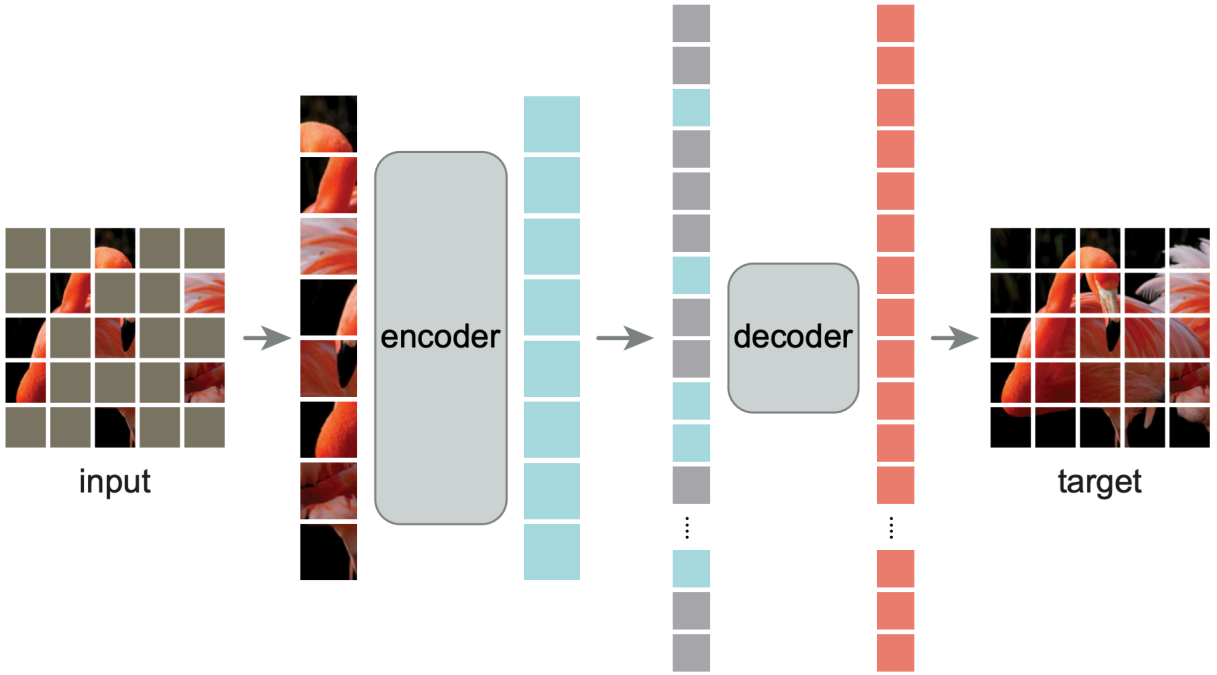
Generative SSL on graphs

- **Generative SSL can naturally avoid the issue of relying on complicated strategies.**
- **But previous generative SSL often fail to catch up with the performance of contrastive methods.**

What makes graph generative learning lag behind contrastive learning ?

Generative SSL has been gaining increasing significance

- MAE[He, 2021] leads the revolution and opens the new era of generative SSL
  - Get rid of all constraints of contrastive SSL



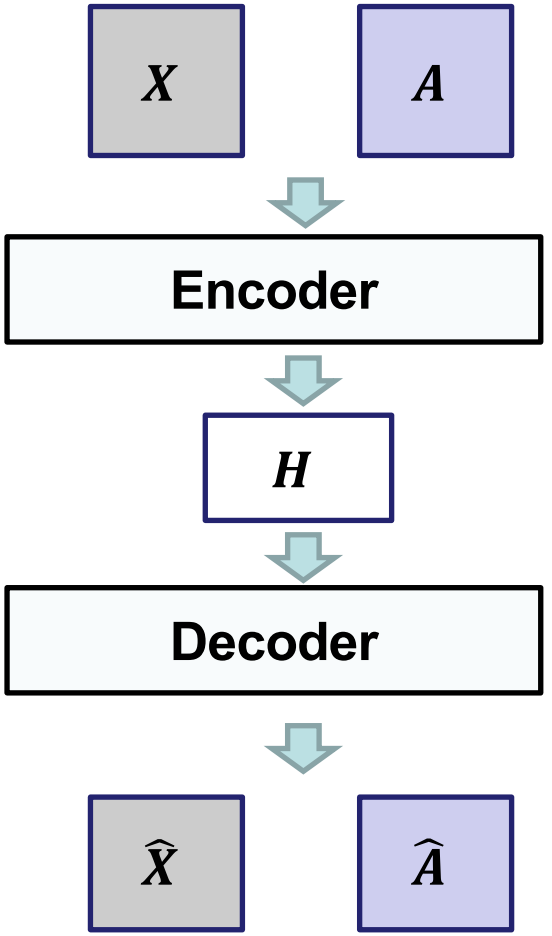
How do we unleash the potential of generative SSL on graphs ?

Problem

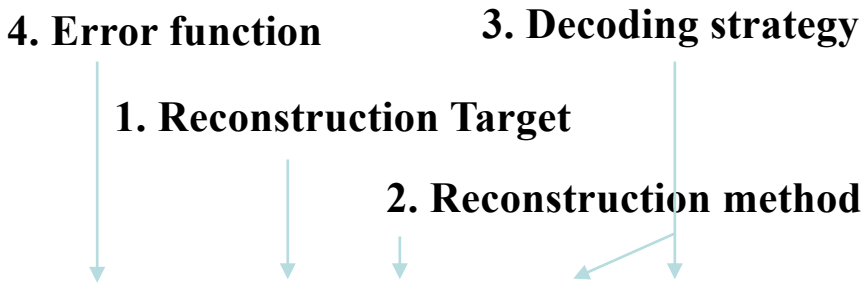
Graph Autoencoders

Graph AutoEncoder

- $G = (V, A, X)$
  - $A \in \{0,1\}^{N \times N}$: adjacency matrix,
  - $X \in \mathbb{R}^{N \times d}$: node features
- Encoding: $H = f_E(A, X)$,
- Decoding: $G' = f_D(A, H)$
- Reconstruction objectives:
  - graph structure (link)
  - Node features



Summary of Graph AutoEncoder

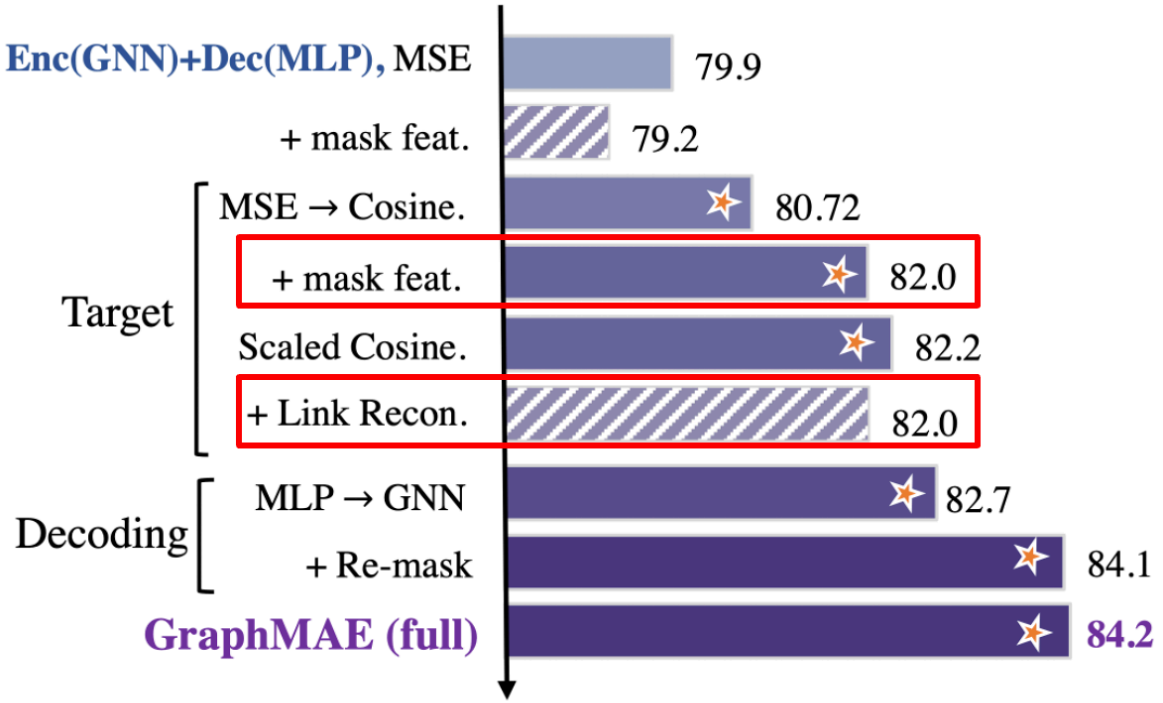


Methods	Feat. Loss	AE	No Struc.	Mask Feat.	GNN Decoder	Re-mask Dec.	Space
VGAE [20]	n/a	✓	-	-	-	-	$O(N^2)$
ARVGA [26]	n/a	✓	-	-	-	-	$O(N^2)$
MGAE [42]	MSE	✓	-	✓	-	-	$O(N)$
GALA [27]	MSE	✓	✓	-	✓	-	$O(N)$
GATE [31]	MSE	✓	-	-	✓	-	$O(N)$
AttrMask [16]	CE	✓	✓	✓	-	-	$O(N)$
GPT-GNN [17]	MSE	-	-	✓	-	-	$O(N)$
AGE [3]	n/a	✓	-	-	-	-	$O(N^2)$
NodeProp [18]	MSE	✓	✓	✓	-	-	$O(N)$
GraphMAE	SCE	✓	✓	✓	✓	✓	$O(N)$

(a) Technical comparison between generative SSL methods.

Critical components

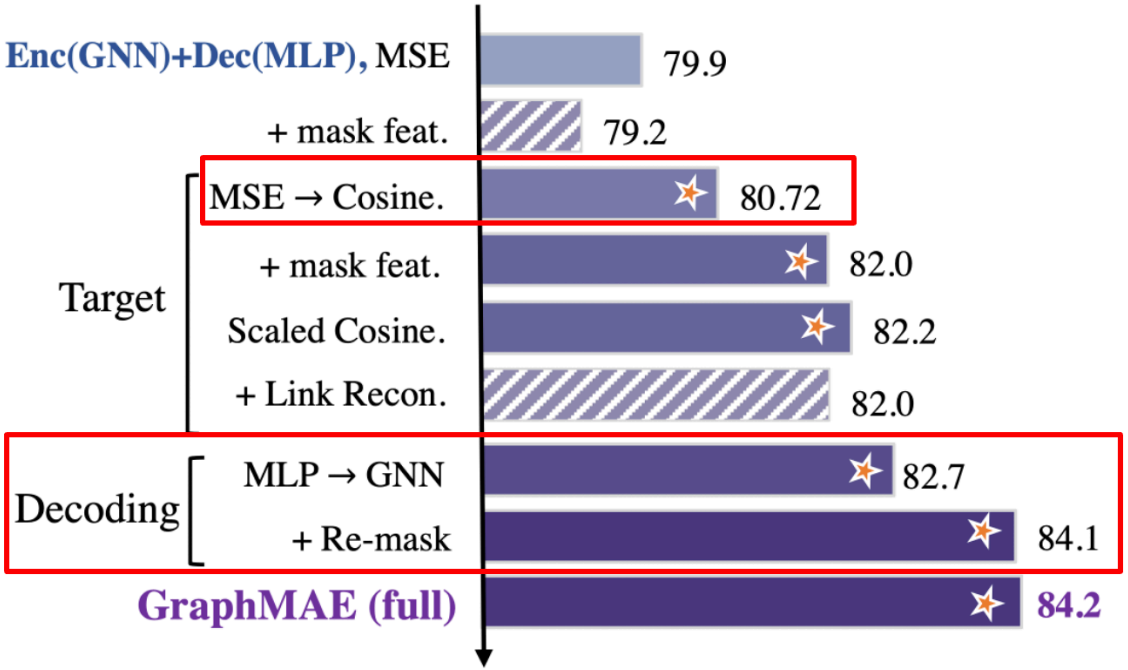
1. **Link reconstruction** may be over-emphasized.
2. **Reconstruction without corruption** may not be robust



(b) The effect of GraphMAE designs on the performance on Cora dataset.

Critical components

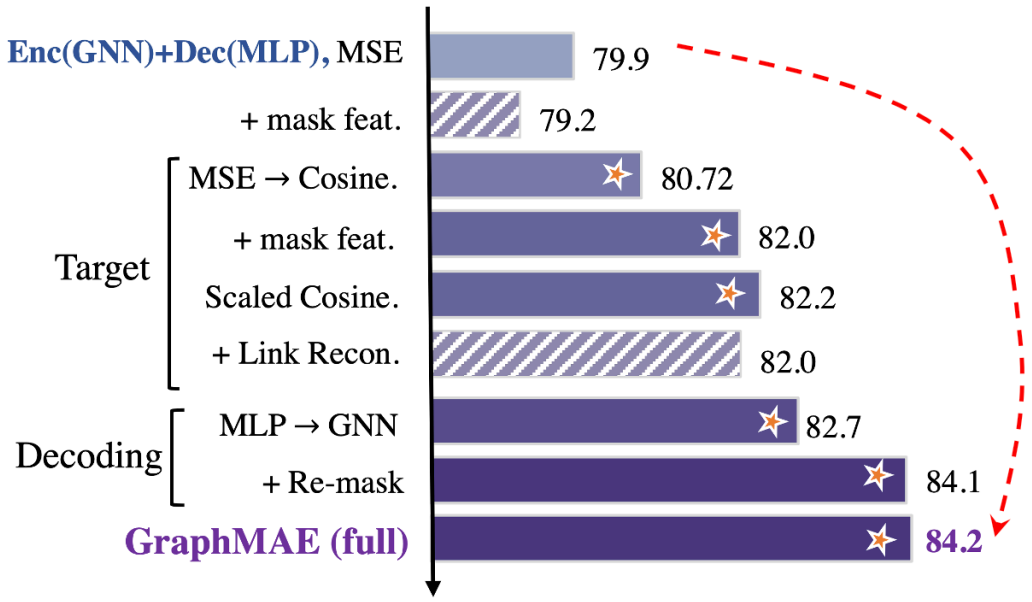
3. **Linear/MLP** is a less expressive decoding strategy
4. **MSE** may not be a good criterion for feature reconstruction in graph



(b) The effect of GraphMAE designs on the performance on Cora dataset.

Generative SSL for graph ?

1. What to reconstruct ?
2. How to avoid trivial solutions ?
3. How to design the decoding ?
4. What error function to use ?



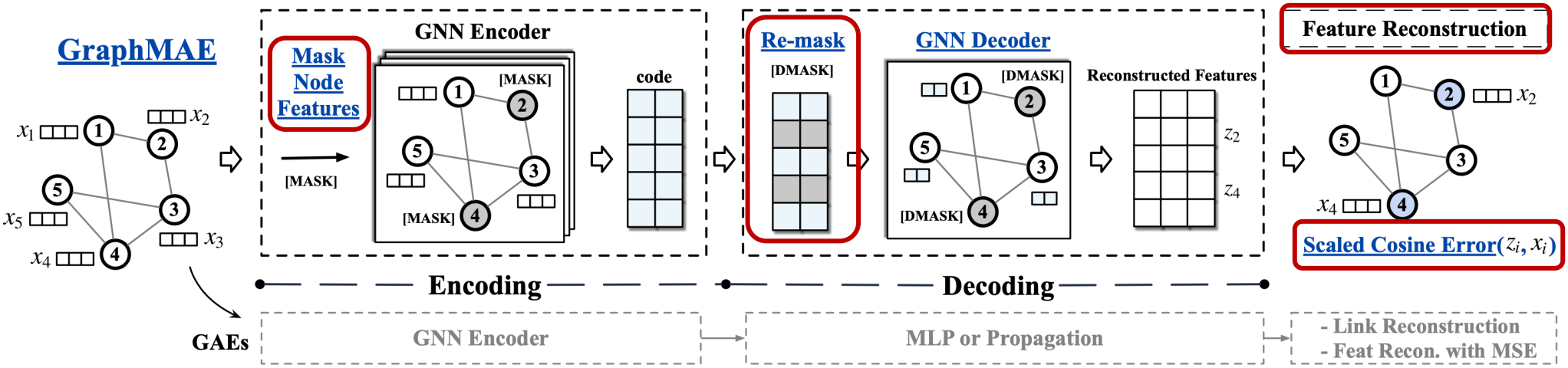
(b) The effect of GraphMAE designs on the performance on Cora dataset.

GraphMAE Framework

Self-Supervised Masked Graph Autoencoders

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-Supervised Masked Graph Autoencoders. In KDD'22.

GraphMAE Method



- Masked feature reconstruction
- GNN as decoder with re-mask decoding
- Scaled cosine error as the Criterion

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-Supervised Masked Graph Autoencoders. In KDD'22.

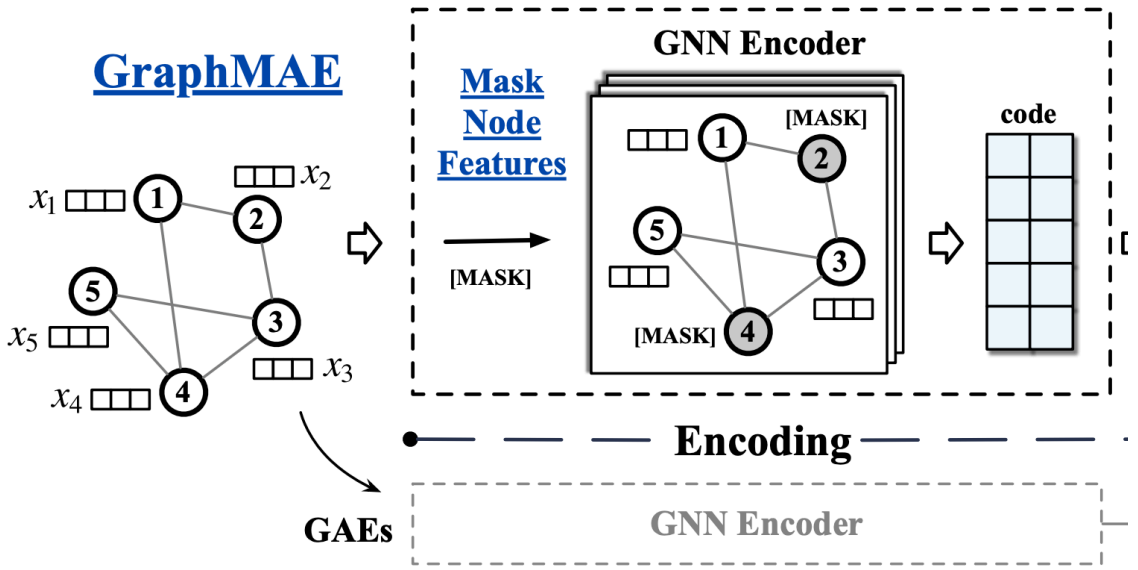
Masked feature reconstruction

- Feature construction as the learning objective
- Masked feature reconstruction
  1. Sample a subset of nodes
     - $\widetilde{V} \subset V$
  2. Replace node feature with [MASK]

$$\widetilde{\boldsymbol{x}}_i = \begin{cases} \boldsymbol{x}_{[M]} & v_i \in \widetilde{\mathcal{V}} \\ \boldsymbol{x}_i & v_i \notin \widetilde{\mathcal{V}} \end{cases}$$

- $H = f_E(A, \tilde{X})$

1. What to reconstruct ?
2. How to avoid trivial solutions ?



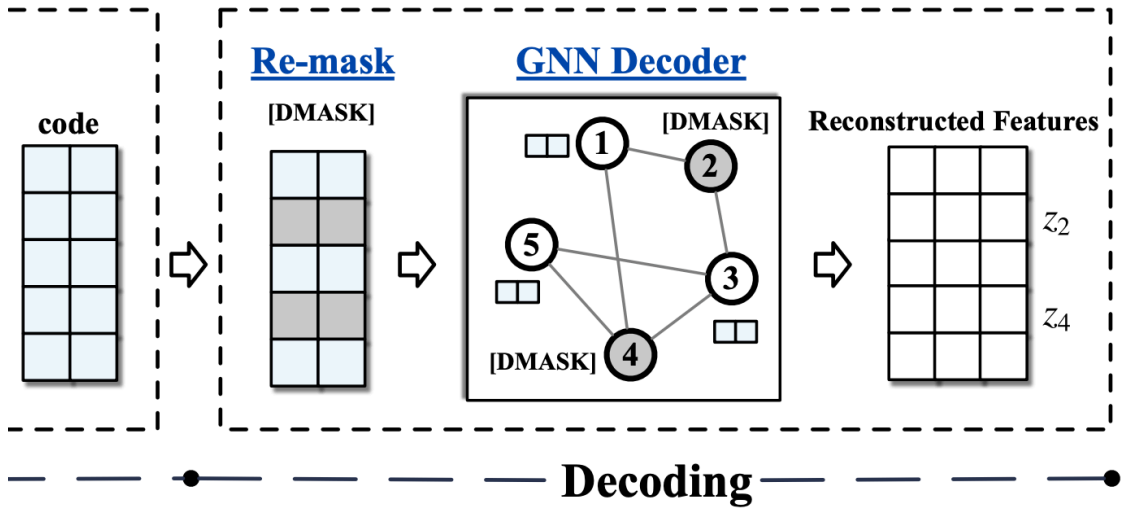
GNN as decoder with re-mask decoding

- Use GNN as the decoder
  - A more expressive decoder helps reconstruct low informative features
- Re-mask node features before decoder
  - Re-mask the "masked" nodes

3. How to design the decoding ?

- $\widetilde{H} = \text{Remask}(H),\ Z = f_D(A, \widetilde{H})$

$$\widetilde{\boldsymbol{h}}_i = \begin{cases} \boldsymbol{h}_{[M]} & v_i \in \widetilde{\mathcal{V}} \\ \boldsymbol{h}_i & v_i \notin \widetilde{\mathcal{V}} \end{cases}$$



Scaled cosine error as the criterion

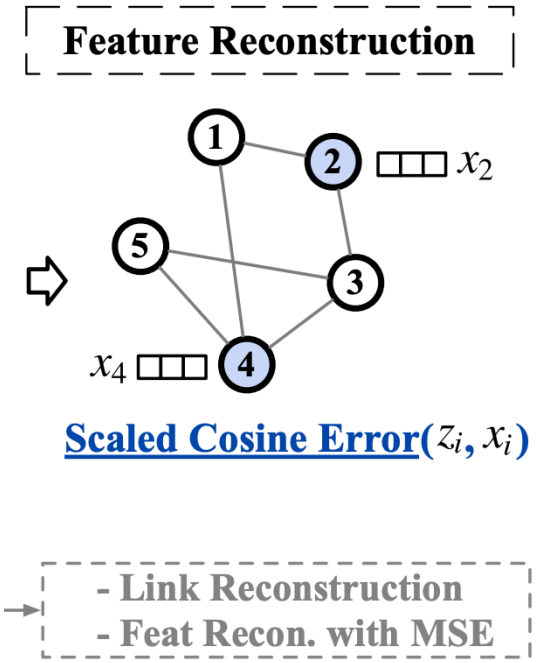
- MSE fails, especially for continuous features
  - *Sensitivity & low selectivity*

$$L_{MSE} = \frac{1}{|\tilde{V}|}\sum_{v_i \in \tilde{V}} (x_i - z_i)^2$$

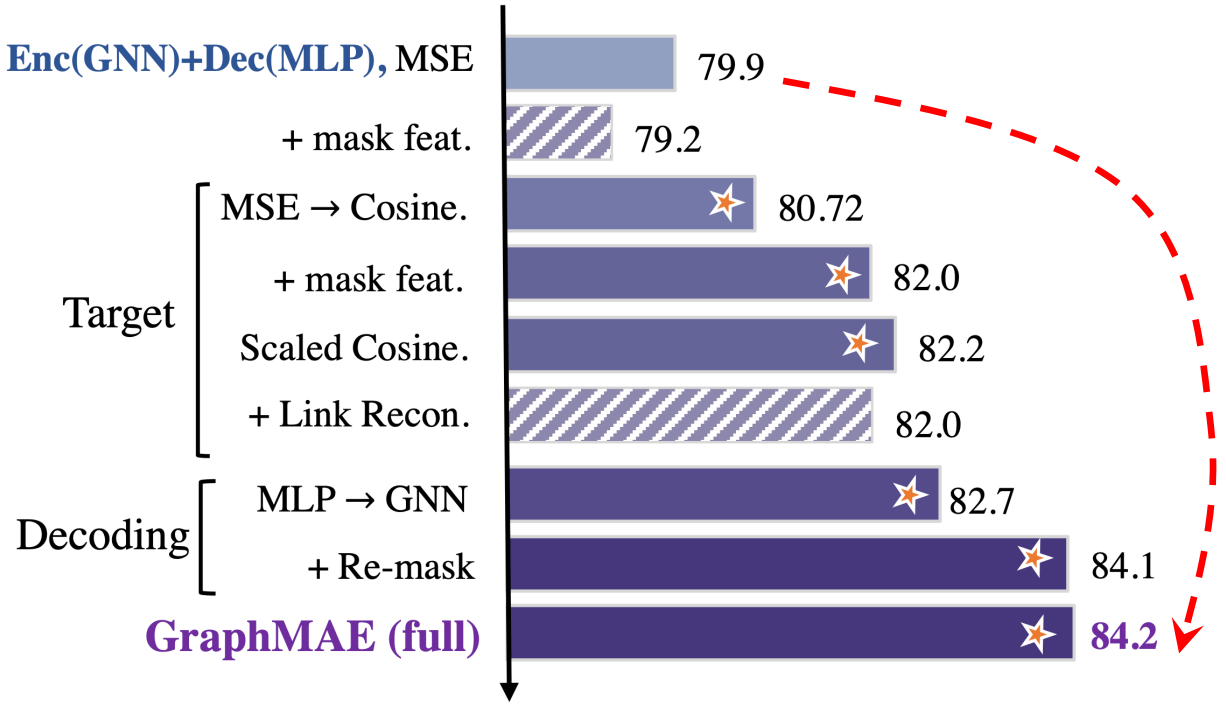
- Scaled cosine error as the criterion.
  - *Cosine error & Scaled coefficient*

$$\mathcal{L}_{\text{SCE}} = \frac{1}{|\widetilde{\mathcal{V}}|}\sum_{v_i \in \widetilde{\mathcal{V}}} \left(1 - \frac{\boldsymbol{x}_i^T \boldsymbol{z}_i}{\|\boldsymbol{x}_i\| \cdot \|\boldsymbol{z}_i\|}\right)^{\gamma},\ \gamma \geq 1,$$

4. What error function to use ?



Overall Improvement of GraphMAE



(b) The effect of GraphMAE designs on the performance on Cora dataset. *Node classification*

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-Supervised Masked Graph Autoencoders. In KDD'22.

Experiments

- Unsupervised representation learning for *Node classification*
- Unsupervised representation learning for *Graph classification*
- *Transfer learning* on molecular property prediction

Code: https://github.com/THUDM/GraphMAE

Node classification

Table 1: Experiment results in unsupervised representation learning for node classification. We report Micro-F1(%) score for PPI and accuracy(%) for the other datasets.

	Dataset	Cora	CiteSeer	PubMed	Ogbn-arxiv	PPI	Reddit
Supervised	GCN	81.5	70.3	79.0	71.74±0.29	75.7±0.1	95.3±0.1
	GAT	83.0±0.7	72.5±0.7	79.0±0.3	72.10±0.13	97.30±0.20	96.0±0.1
Self-supervised	GAE	71.5±0.4	65.8±0.4	72.1±0.5	-	-	-
	GPT-GNN	80.1±1.0	68.4±1.6	76.3±0.8	-	-	-
	GATE	83.2±0.6	71.8±0.8	80.9±0.3	-	-	-
Contrastive methods	DGI	82.3±0.6	71.8±0.7	76.8±0.6	70.34±0.16	63.80±0.20	94.0±0.10
	MVGRL	83.5±0.4	73.3±0.5	80.1±0.7	-	-	-
	GRACE[1]	81.9±0.4	71.2±0.5	80.6±0.4	71.51±0.11	69.71±0.17	94.72±0.04
	BGRL[1]	82.7±0.6	71.1±0.8	79.6±0.5	71.64±0.12	73.63±0.16	94.22±0.03
	InfoGCL	83.5±0.3	**73.5±0.4**	79.1±0.2	-	-	-
	CCA-SSG[1]	84.0±0.4	73.1±0.3	81.0±0.4	71.24±0.20	73.34±0.17	95.07±0.02
	GraphMAE	**84.2±0.4**	73.4±0.4	**81.1±0.4**	**71.75±0.17**	**74.50±0.29**	**96.01±0.08**

Code: https://github.com/THUDM/GraphMAE

Graph classification

Table 2: Experiment results in unsupervised representation learning for graph classification. We report accuracy(%) for all datasets.

	Dataset	IMDB-B	IMDB-M	PROTEINS	COLLAB	MUTAG	REDDIT-B	NCI1
Supervised	GIN	75.1±5.1	52.3±2.8	76.2±2.8	80.2±1.9	89.4±5.6	92.4±2.5	82.7±1.7
	DiffPool	72.6±3.9	-	75.1±3.5	78.9±2.3	85.0±10.3	92.1±2.6	-
Graph Kernels	WL	72.30±3.44	46.95±0.46	72.92±0.56	-	80.72±3.00	68.82±0.41	80.31±0.46
	DGK	66.96±0.56	44.55±0.52	73.30±0.82	-	87.44±2.72	78.04±0.39	80.31±0.46
Self-supervised	graph2vec	71.10±0.54	50.44±0.87	73.30±2.05	-	83.15±9.25	75.78±1.03	73.22±1.81
	Infograph	73.03±0.87	49.69±0.53	74.44±0.31	70.65±1.13	89.01±1.13	82.50±1.42	76.20±1.06
	GraphCL	71.14±0.44	48.58±0.67	74.39±0.45	71.36±1.15	86.80±1.34	89.53±0.84	77.87±0.41
	JOAO	70.21±3.08	49.20±0.77	74.55±0.41	69.50±0.36	87.35±1.02	85.29±1.35	78.07±0.47
	GCC	72.0	49.4	-	78.9	-	**89.8**	-
	MVGRL	74.20±0.70	51.20±0.50	-	-	89.70±1.10	84.50±0.60	-
	InfoGCL	75.10±0.90	51.40±0.80	-	80.00±1.30	**91.20±1.30**	-	80.20±0.60
	GraphMAE	**75.52±0.66**	**51.63±0.52**	**75.30±0.39**	**80.32±0.46**	88.19±1.26	88.01±0.19	**80.40±0.30**

Contrastive methods

Code: https://github.com/THUDM/GraphMAE

Transfer learning

Table 3: Experiment results in transfer learning on molecular property prediction benchmarks. The model is first pre-trained on ZINC15 and then finetuned on the following datasets. We report ROC-AUC(%) scores.

Contrastive methods

	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	Avg.
No-pretrain	65.5±1.8	74.3±0.5	63.3±1.5	57.2±0.7	58.2±2.8	71.7±2.3	75.4±1.5	70.0±2.5	67.0
ContextPred	64.3±2.8	75.7±0.7	63.9±0.6	60.9±0.6	65.9±3.8	75.8±1.7	77.3±1.0	79.6±1.2	70.4
AttrMasking	64.3±2.8	**76.7±0.4**	**64.2±0.5**	61.0±0.7	71.8±4.1	74.7±1.4	77.2±1.1	79.3±1.6	71.1
Infomax	68.8 ±0.8	75.3 ±0.5	62.7 ±0.4	58.4 ±0.8	69.9±3.0	75.3 ±2.5	76.0 ±0.7	75.9 ±1.6	70.3
GraphCL	69.7±0.7	73.9±0.7	62.4±0.6	60.5±0.9	76.0±2.7	69.8±2.7	**78.5±1.2**	75.4±1.4	70.8
JOAO	70.2±1.0	75.0±0.3	62.9±0.5	60.0±0.8	81.3±2.5	71.7±1.4	76.7±1.2	77.3±0.5	71.9
GraphLoG	**72.5±0.8**	75.7±0.5	63.5±0.7	**61.2±1.1**	76.7±3.3	76.0±1.1	77.8±0.8	**83.5±1.2**	73.4
GraphMAE	72.0±0.6	75.5±0.6	64.1±0.3	60.3±1.1	**82.3±1.2**	**76.3±2.4**	77.2±1.0	83.1±0.9	**73.8**

Code: https://github.com/THUDM/GraphMAE

Ablation study

- Effect of decoder type, objective function and mask strategy

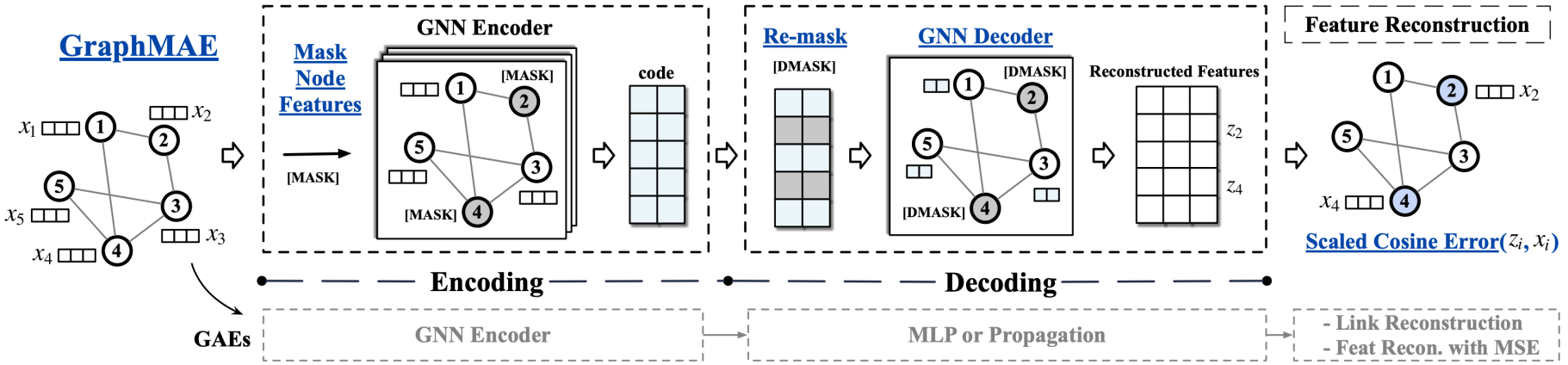
Table 4: Ablation studies of decoder type, re-mask and reconstruction criterion on node- and graph-level benchmarks.

	Dataset	Node-Level			Graph-Level	
		Cora	PubMed	Arxiv	MUTAG	IMDB-B
COMP.	GraphMAE	84.2	81.1	71.75	88.19	75.52
	w/o mask	79.7	77.9	70.97	82.58	74.42
	w/o re-mask	82.7	80.0	71.61	86.29	74.42
	w/ MSE	79.1	73.1	67.44	86.30	74.04
Decoder	MLP	82.2	80.4	71.54	87.16	73.94
	GCN	81.3	79.1	71.59	87.78	74.54
	GIN	81.8	80.2	71.41	88.19	75.52
	GAT	84.2	81.1	71.75	86.27	74.04

Code: https://github.com/THUDM/GraphMAE

Summary

- Explore generative self-supervised learning in graphs
- Identify the common issues in current graph autoencoders.
- Present a simple and improved masked autoencoder—GraphMAE
- The experimental results show that generative SSL can have great potential



Code: https://github.com/THUDM/GraphMAE

Summary

- GCC: Graph Contrastive Coding for GNN Pre-Training
  - Motivation: universal structural patterns across networks?
  - Pre-training on the graph structure via contrastive learning
  - Fine-tuning on different downstream graph tasks

- GraphMAE: Self-Supervised Masked Graph Autoencoders
  - Motivation: why graph generative learning lag behind graph contrastive learning?
  - Target generative self-supervised learning on graphs
  - Based on a simple architecture: masked auto-encoder

Homework 7: Graph SSL & Pre-training

- Comment on graph SSL & pre-training:
  - Due by 4th Sept.
  - **No coding**
  - Write your comments (Reading other papers if possible)
  - Post them to https://discuss.cogdl.ai/t/topic/96
  - Discuss with others
  - Send your comments and discussions (via screenshots) to our email

- Reminder: Homework 1~6 & course proposal

Thank you !

Collaborators:

Zhenyu Hou, Yuxiao Dong, Jie Tang, et al. (**THU**)

Qingfei Zhao, Xinije Zhang, Peng Zhang, et al. (**Zhipu AI**)

Hongxiao Yang, Chang Zhou, et al. (**Alibaba**)

Yang Yang (**ZJU**)

Yukuo Cen, KEG, Tsinghua U.

https://github.com/THUDM/cogdl

Online Discussion Forum https://discuss.cogdl.ai/