Self-supervised Learning and Pre-training on Graphs (GNNs)

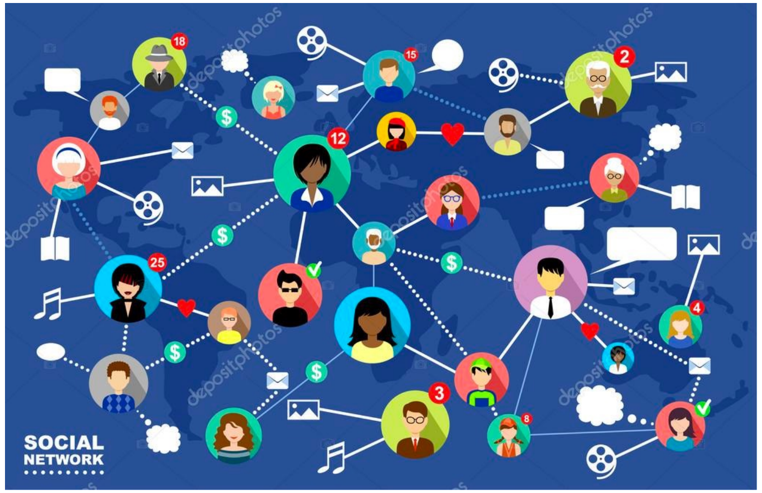
Jie Tang

KEG, Tsinghua University

Download the slides here

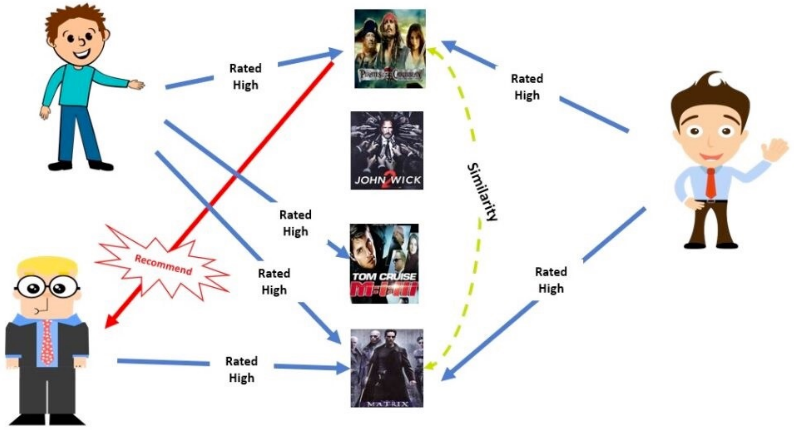
Graph

- Graph data exists everywhere



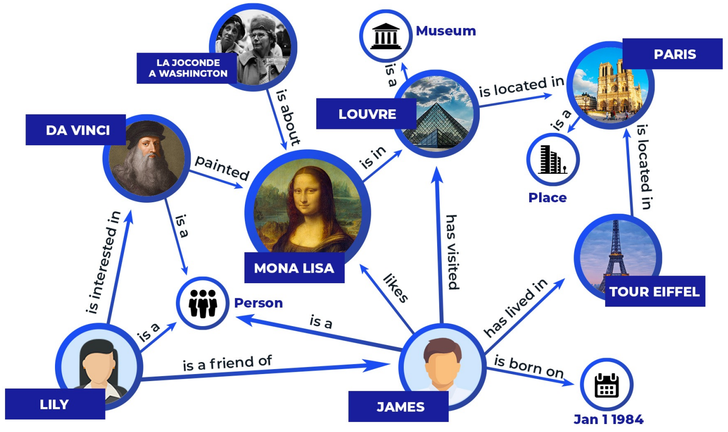
Social Network

- WeChat: 1.2 billion users
  61 billion links



Recommender System

- Alibaba: 2.3 billion trans. on 11/11



Knowledge Graph

- Wikidata: >1.4 billion triples

*"The number of **graph neural network** papers in this journal has grown as the field matures. We take a closer look at some of the **scientific applications**."*

1. The graph connection. *Nature Machine Intelligence* **4**, 187–188 (2022). https://doi.org/10.1038/s42256-022-00476-6

Machine Learning on Graphs

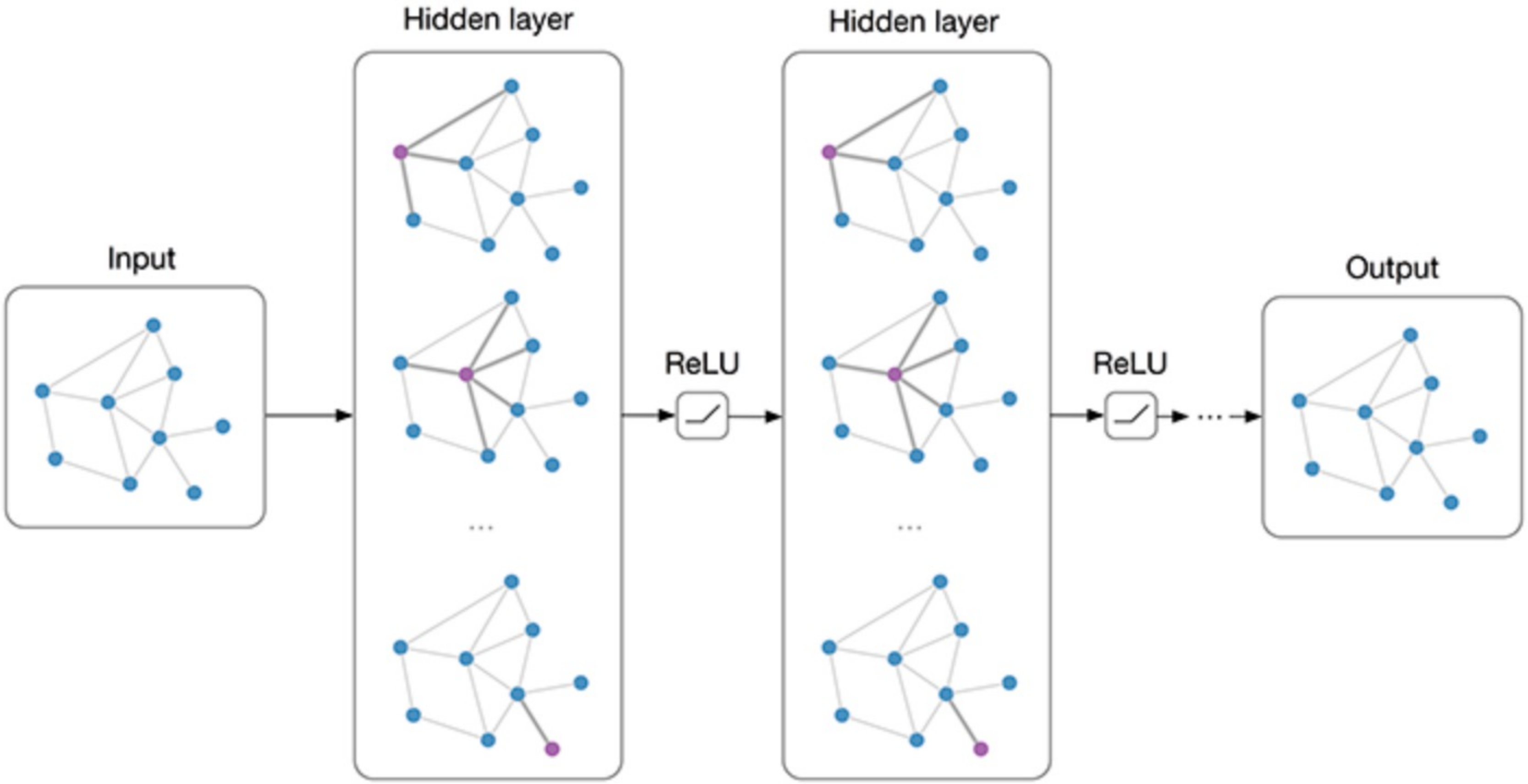
- ML tasks on Graphs:
  - Node classification
    - Predict a type of a given node
  - Link prediction
    - Predict whether two nodes are linked
  - Graph classification
    - Predict the properties of molecules
  - Community detection
    - Identify densely linked clusters of nodes

Learning on Graphs with Graph Neural Networks (GNNs)

- A question: Are you using GNNs?

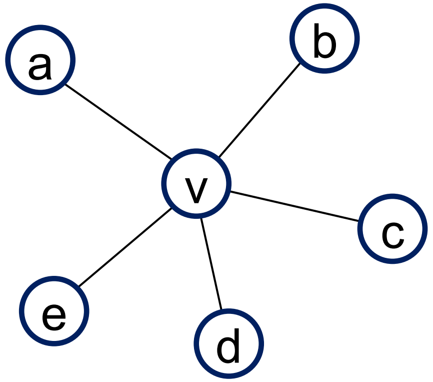
Graph Neural Networks

- Layer-wise propagation: $$f(H^{(l)}, A) = \sigma\left(AH^{(l)}W^{(l)}\right)$$



Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In ICLR ’17.

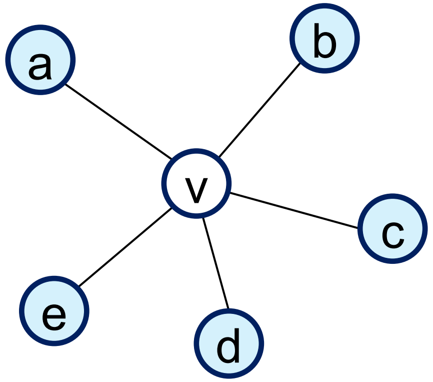
Graph Neural Networks



$$\boldsymbol{h}_v = f(\boldsymbol{h}_a, \boldsymbol{h}_b, \boldsymbol{h}_c, \boldsymbol{h}_d, \boldsymbol{h}_e)$$

- **Neighborhood Aggregation:**
  - Aggregate neighbor information and pass into a neural network
  - It can be viewed as a center-surround filter in CNN---graph convolutions!

GCN: Graph Convolutional Networks



parameters in layer k

Non-linear activation function (e.g., ReLU)

$$\boldsymbol{h}_v^k = \sigma\left(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{h_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

node v's embedding at layer k

the neighbors of node v

1. Kipf et al. Semi-supervised Classification with Graph Convolutional Networks. ICLR 2017

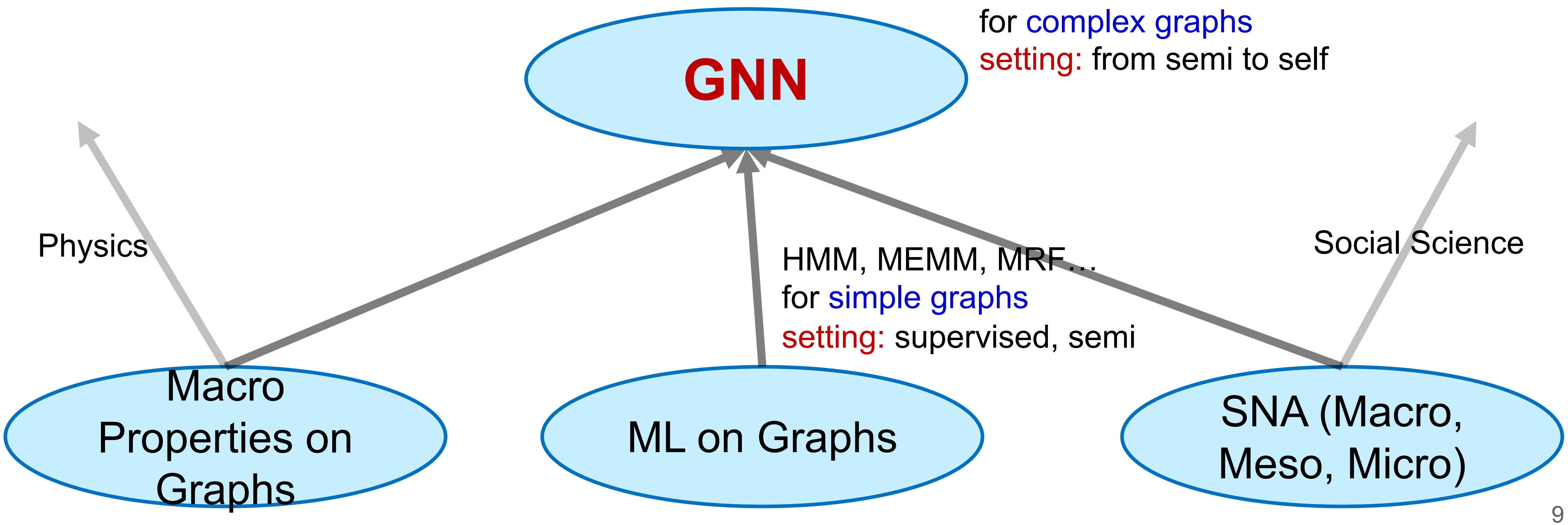
GCN Performance

- 2-layer GCN： $\boldsymbol{Z} = \mathrm{softmax}\left(\widetilde{\boldsymbol{A}}\,\sigma\left(\widetilde{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_0\right)\boldsymbol{W}_1\right)$

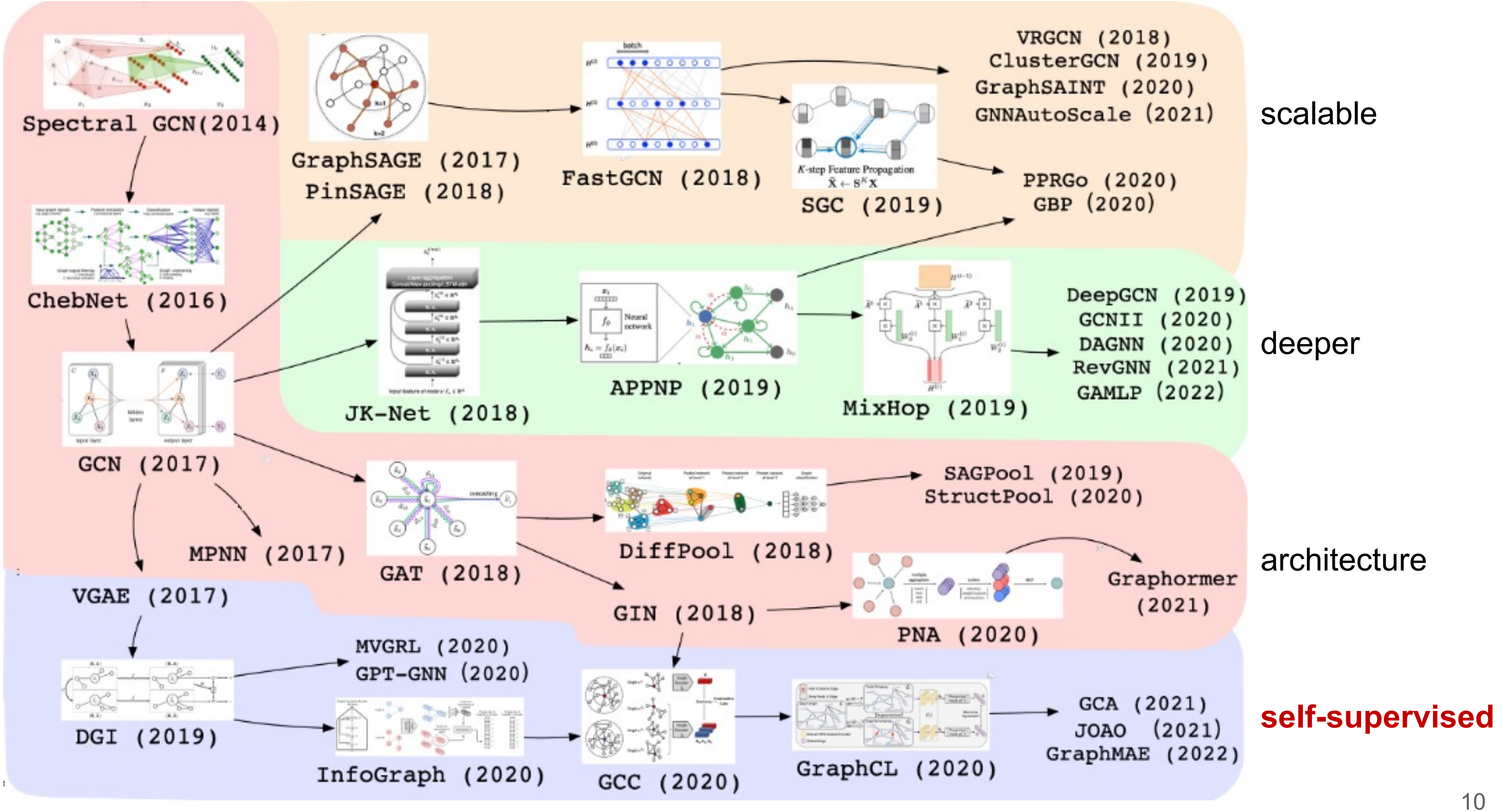
Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)

Why GNN?


GNN
for complex graphs
setting: from semi to self
Physics
HMM, MEMM, MRF…
for simple graphs
setting: supervised, semi
Social Science
Macro Properties on Graphs
ML on Graphs
SNA (Macro, Meso, Micro)

GNN History



Spectral GCN(2014)
ChebNet (2016)
GCN (2017)
MPNN (2017)
VGAE (2017)
DGI (2019)
GraphSAGE (2017)
PinSAGE (2018)
FastGCN (2018)
SGC (2019)
K-step Feature Propagation
VRGCN (2018)
ClusterGCN (2019)
GraphSAINT (2020)
GNNAutoScale (2021)
PPRGo (2020)
GBP (2020)
scalable
JK-Net (2018)
APPNP (2019)
MixHop (2019)
DeepGCN (2019)
GCNII (2020)
DAGNN (2020)
RevGNN (2021)
GAMLP (2022)
deeper
GAT (2018)
DiffPool (2018)
SAGPool (2019)
StructPool (2020)
GIN (2018)
PNA (2020)
Graphormer (2021)
architecture
MVGRL (2020)
GPT-GNN (2020)
InfoGraph (2020)
GCC (2020)
GraphCL (2020)
GCA (2021)
JOAO (2021)
GraphMAE (2022)
self-supervised

Do we really make big progress?

- Using "heterogeneous graph neural networks (HGNN)" as an example
- Unrobust results with biased setting on small data

	HAN [36]		GTN [43]			RSHN [45]			HetGNN [44]				MAGNN [12]	
Dataset	ACM		DBLP	ACM	IMDB	AIFB	MUTAG	BGS	MC (10%)		MC (30%)		DBLP	
Metric	Macro-F1	Micro-F1	Macro-F1	Macro-F1	Macro-F1	Accuracy	Accuracy	Accuracy	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
model*	91.89	91.85	94.18	92.68	60.92	97.22	82.35	93.10	97.8	97.9	98.1	98.2	93.13	93.61
GCN*	89.31	89.45	87.30	91.60	56.89	-	-	-	-	-	-	-	88.00	88.51
GAT*	90.55	90.55	93.71	92.33	58.14	91.67	72.06	66.32	96.2	96.3	96.5	96.5	91.05	91.61
model														
GCN														
GAT														

We tested 12 HGNN algorithms

* With a fairly proper setting, the results are even reversed!

1. Lv et al. Are we really making much progress? Revisiting, benchmarking and refining the Heterogeneous Graph Neural Networks. KDD'21.

Challenges

- Challenge 1: Self-supervised
- Challenge 2: General
- Challenge 3: Robustness

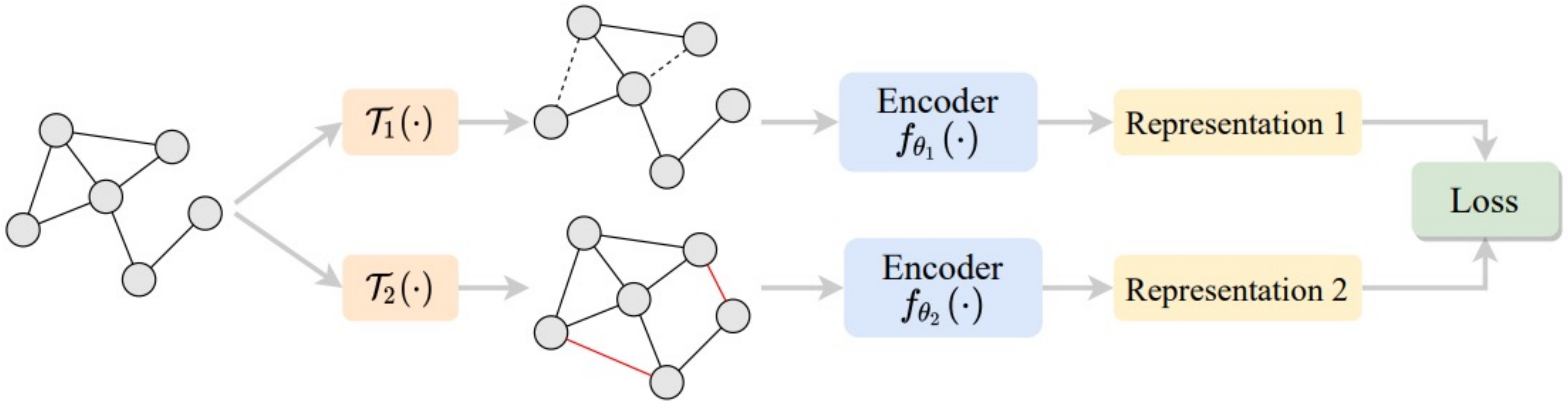
Overview

- Contrastive Self-supervised Learning on Graphs
  - –Training data: all data is unlabeled
  - –Contrasts the generated views
- Generative Self-supervised Learning on Graphs
  - –Training data: all data is unlabeled
  - –Reconstruction of the input graph

Contrastive Learning on Graphs

Paradigm of Graph Contrastive Learning

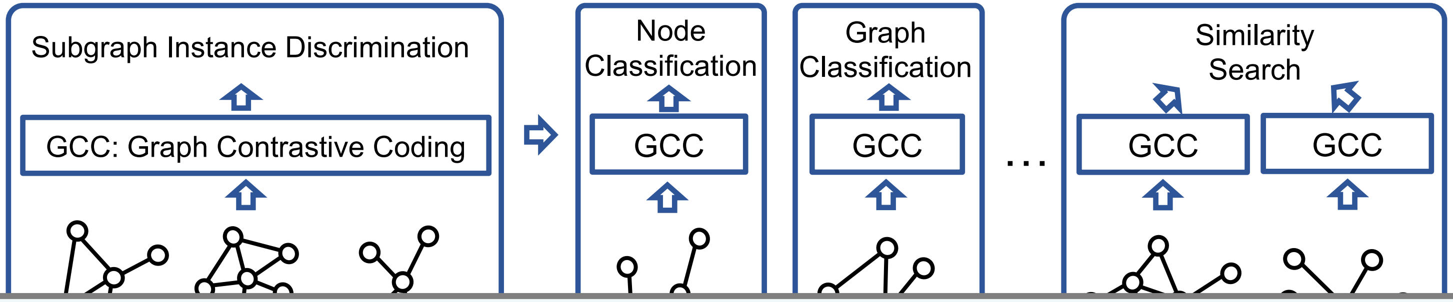
- The graph contrastive learning
  - 1) generates two views based on different augmentations
  - 2) encodes the graphs of two views
  - 3) construct the self-supervised signal via contrast



[1] Wu, Lirong, et al. "Self-supervised learning on graphs: Contrastive, generative, or predictive." IEEE Transactions on Knowledge and Data Engineering (2021).

Graph Contrastive Coding (GCC)

- 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.



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

[1] Jiezhong Qiu et al. 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

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

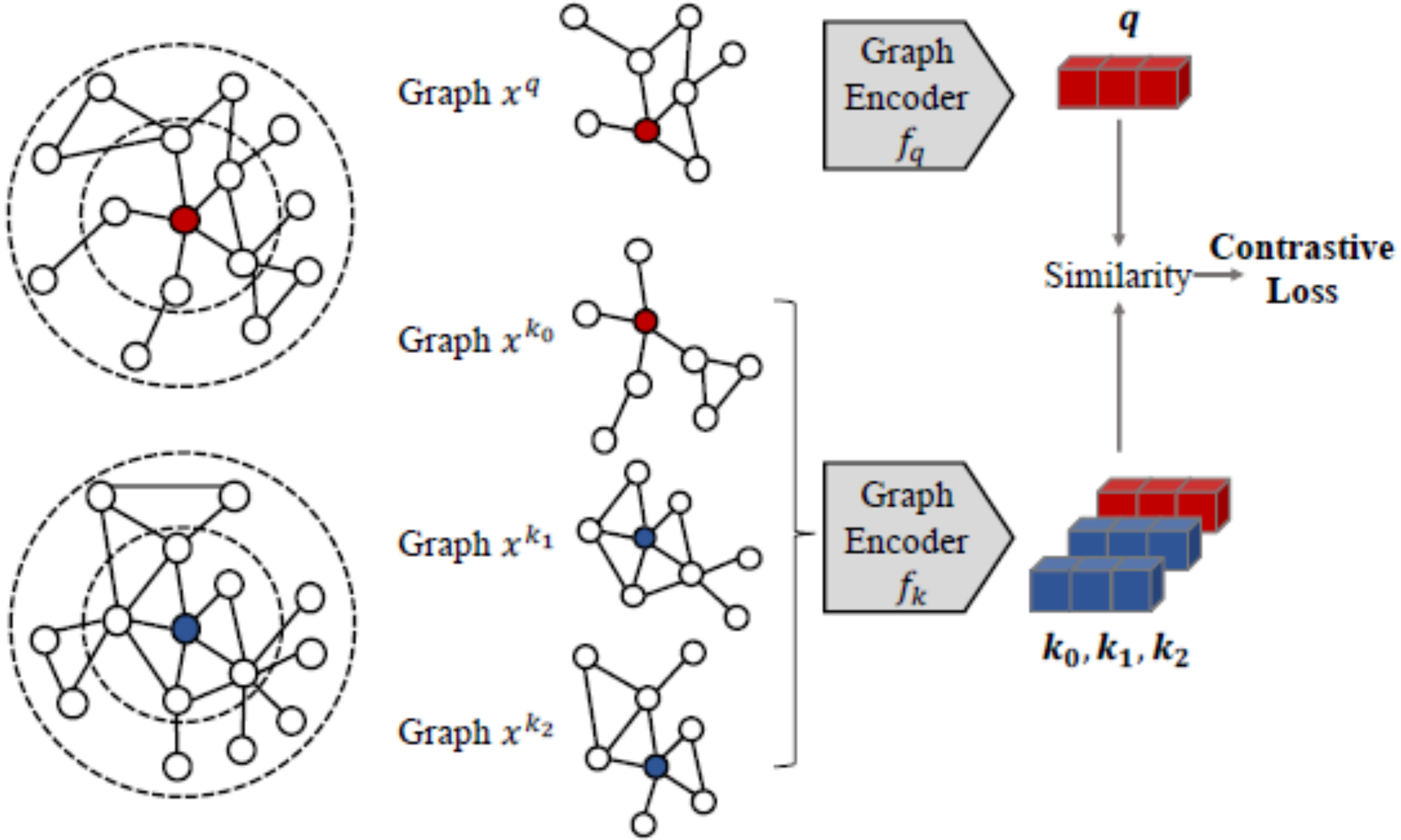
- query instance x^q
- query $\boldsymbol{q}$ (embedding of x^q), i.e., $\boldsymbol{q} = f(x^q)$
- dictionary of keys $\{\boldsymbol{k}_0, \boldsymbol{k}_1, \cdots, \boldsymbol{k}_K\}$
- key $\boldsymbol{k} = f(x^k)$

- 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

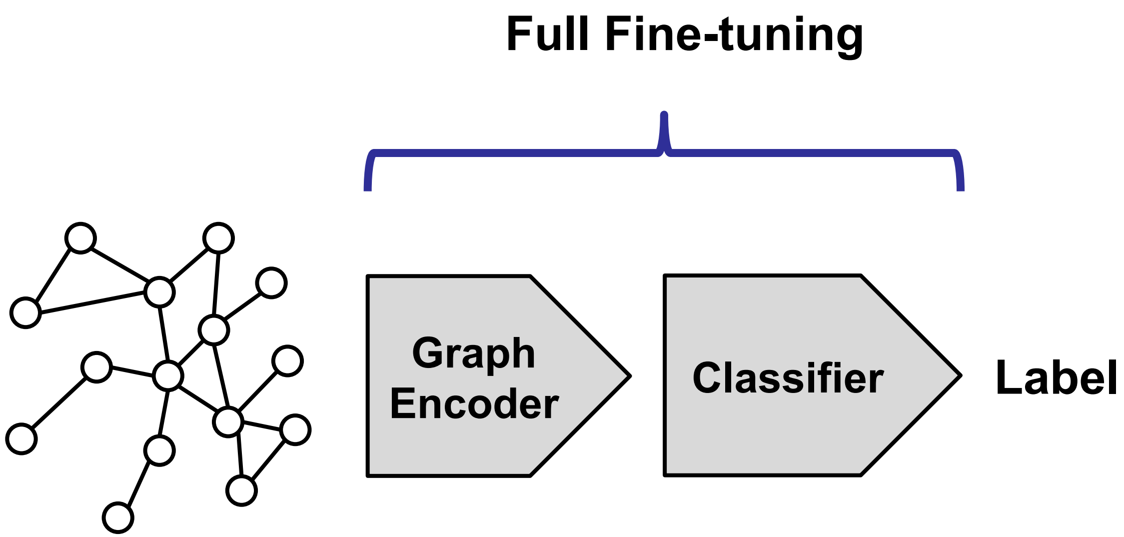
- **Q1**: How to define **instances** in graphs?
- **Q2**: How to define **(dis) similar instance**?
- **Q3**: What are the proper **encoders**?

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

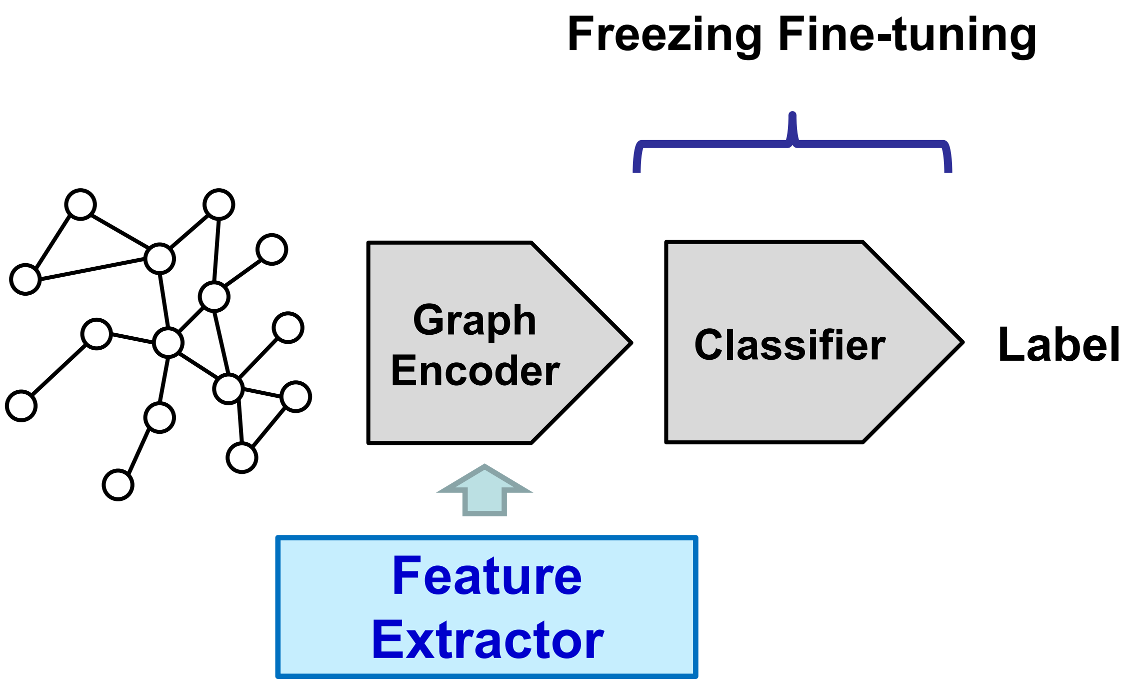


GCC Fine-tuning: Full v.s. Freezing

Full fine-tuning



Freezing fine-tuning

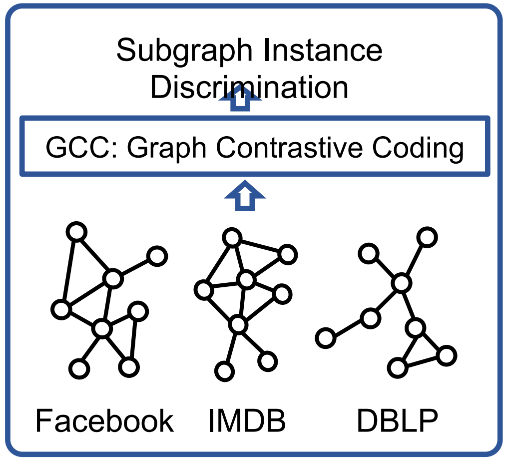


GCC Pre-Training / Fine-tuning

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

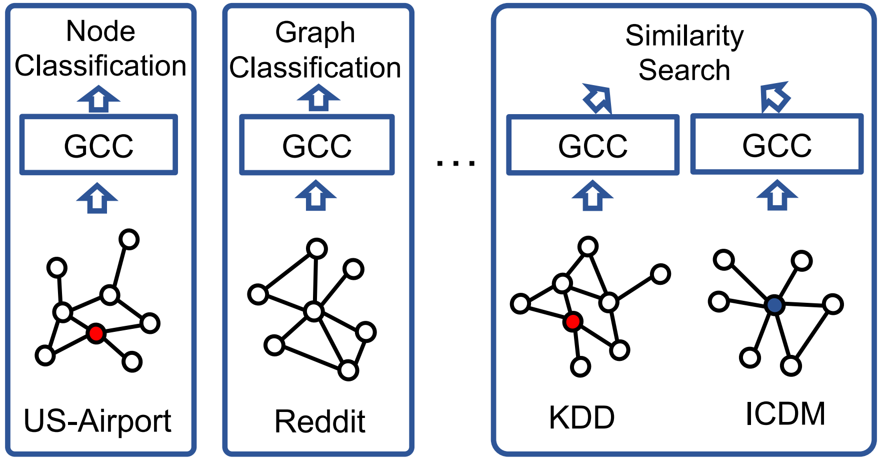
Table 1: Datasets for pre-training, sorted by number of vertices.

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



Pre-Training

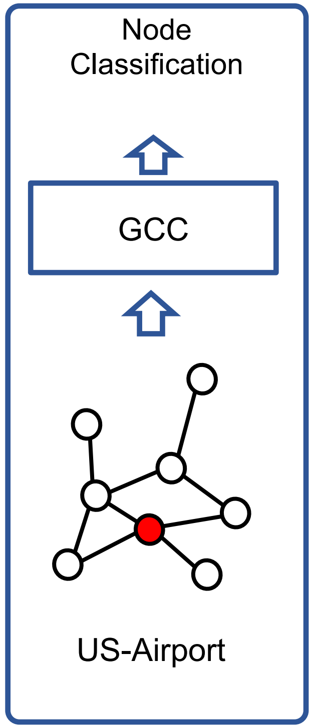
- Fine-tuning Tasks:
  - Node classification
  - Graph classification
  - Top-k Similarity search



Fine-Tuning

Task 1: Node Classification

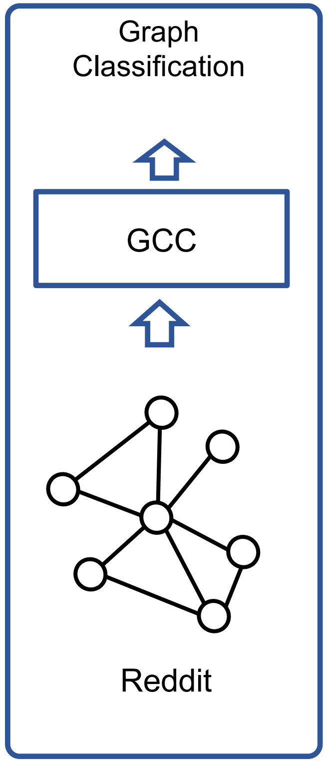
- Setup
  - US-Airport
  - AMiner academic graph



Datasets	US-Airport	H-index
$\lvert V \rvert$	1,190	5,000
$\lvert E \rvert$	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

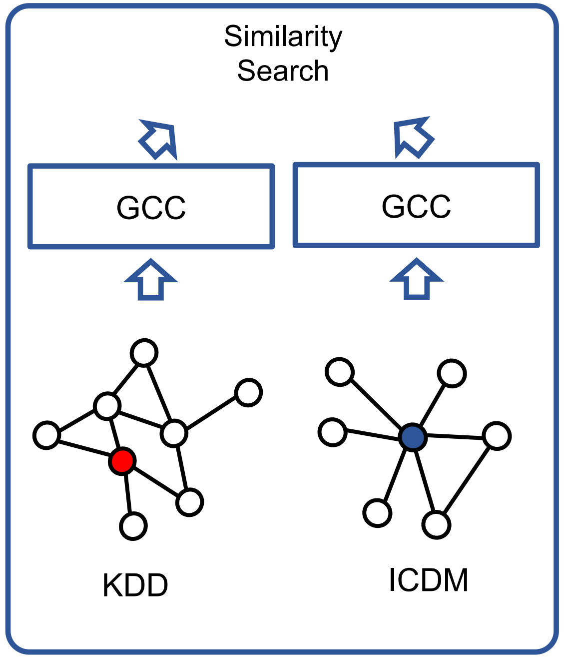
- Setup
  - COLLAB, RDT-B, RDT-M, & IMDB-B, IMDB-M



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

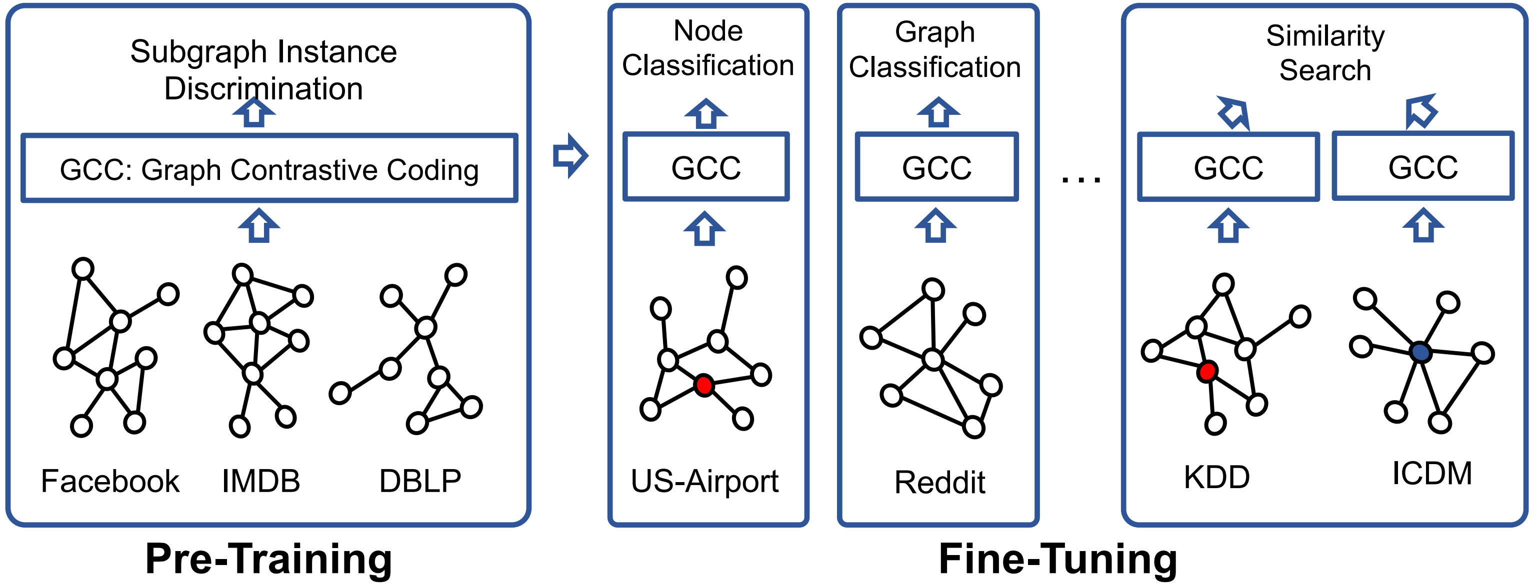
Task 3: Top-k Similarity Search

- 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

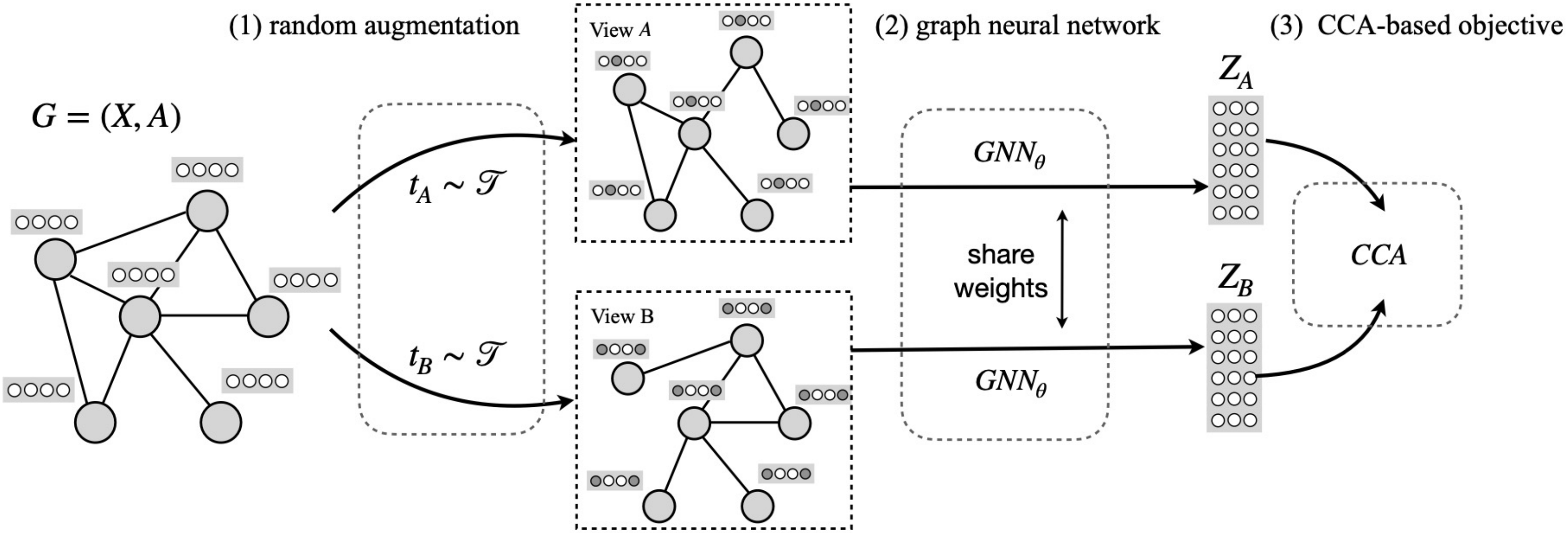
Summary of GCC



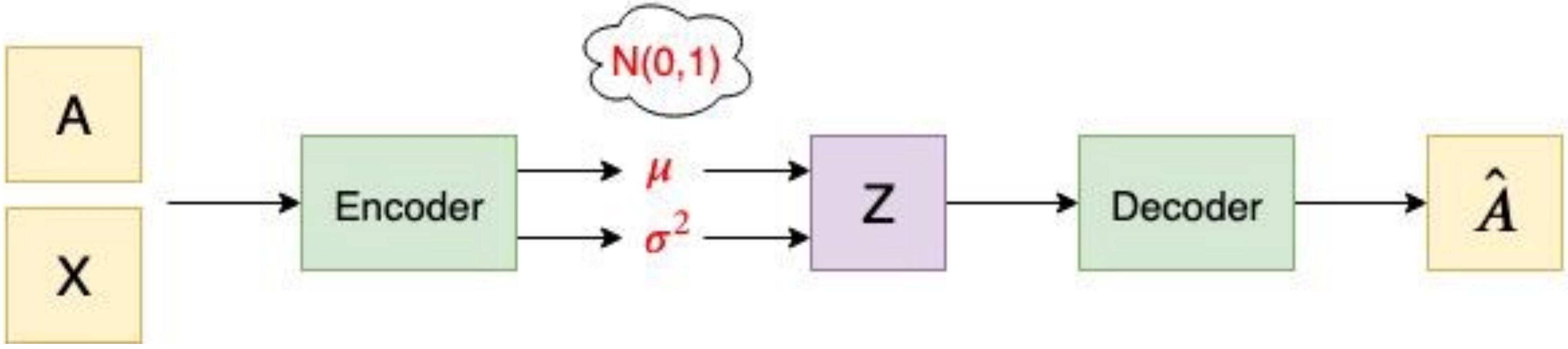
- Pre-training of GNN with transferring structural representations
- Graph Contrastive Coding (GCC) is a graph-based contrastive learning framework for pre-training GNN
- GCC achieves competitive performance to its supervised (trained-from-scratch) counterparts in 3 graph learning tasks on 10 graph datasets.

Generative Learning on Graphs

Self-supervised Learning on Graphs



Contrastive SSL



Generative SSL
(Graph Autoencoder)

Self-supervised Learning on Graphs

- Contrastive SSL has been the dominant approach 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

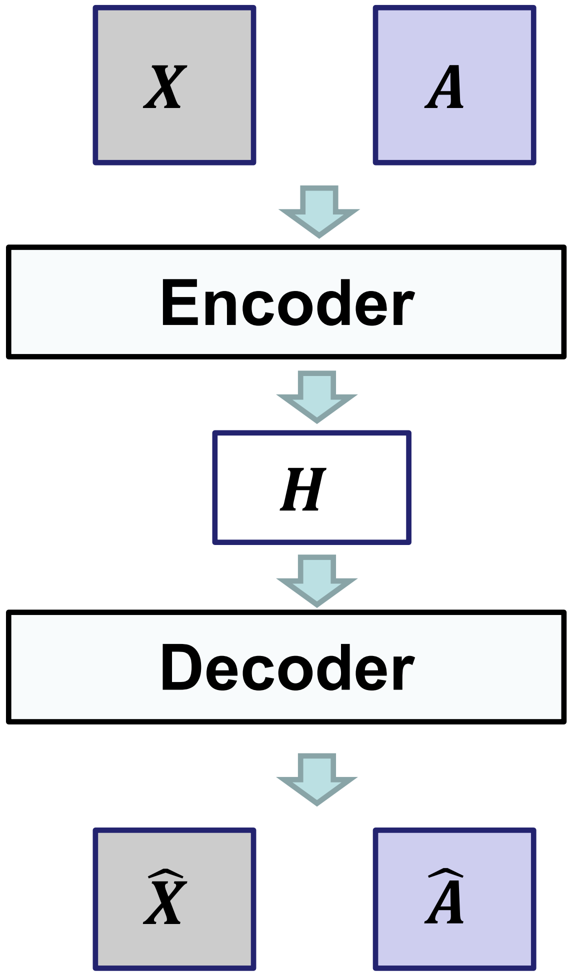
Self-supervised Learning on Graphs

- Contrastive learning heavily 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
      - Momentum-update(BGRL), parameter-noise (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

GraphMAE: A 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



Reconstruction Target

Decoding Strategy

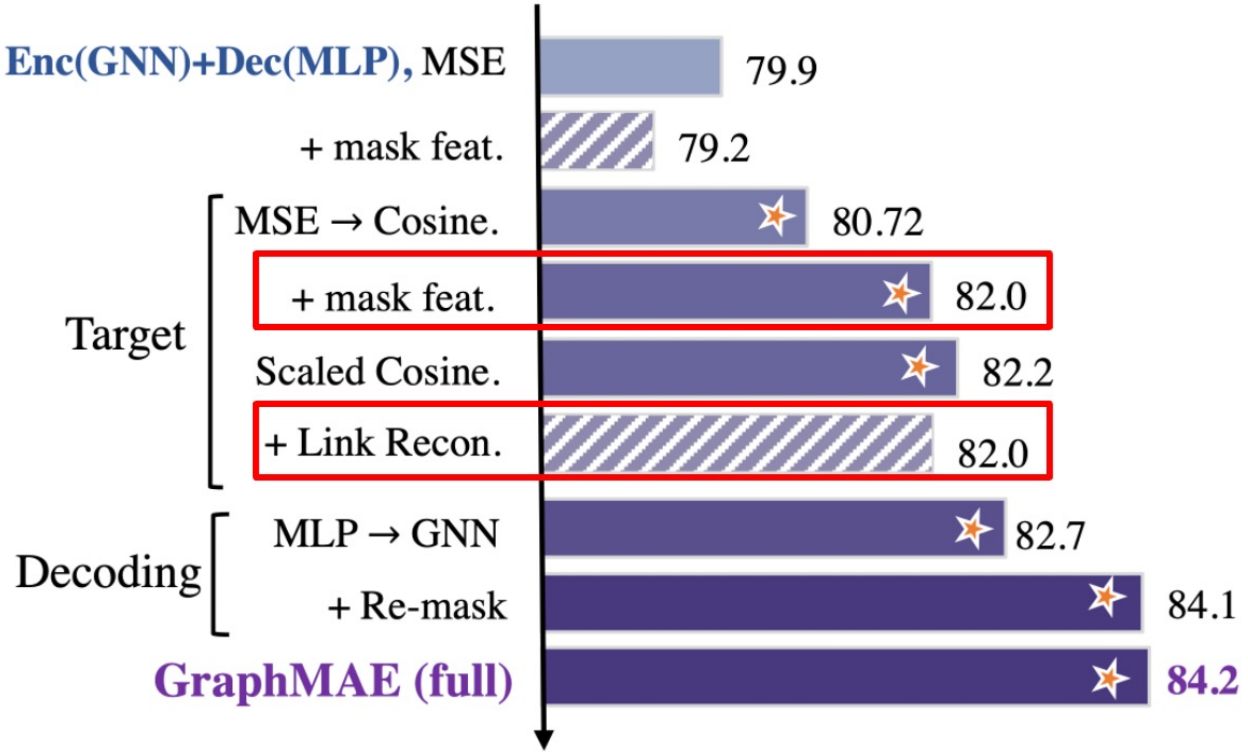
Methods	Feat. Loss	AE	No Struc.	Mask Feat.	GNN Decoder	Re-mask Dec.	Space
VGAE [20]	n/a	✓	-	-	-	-	$O(N^2)$
ARVGA [26]	n/a	✓	-	-	-	-	$O(N^2)$
MGAE [42]	MSE	✓	-	✓	-	-	$O(N)$
GALA [27]	MSE	✓	✓	-	✓	-	$O(N)$
GATE [31]	MSE	✓	-	-	✓	-	$O(N)$
AttrMask [16]	CE	✓	✓	✓	-	-	$O(N)$
GPT-GNN [17]	MSE	-	-	✓	-	-	$O(N)$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

Error Function

Reconstruction Method

Critical Components

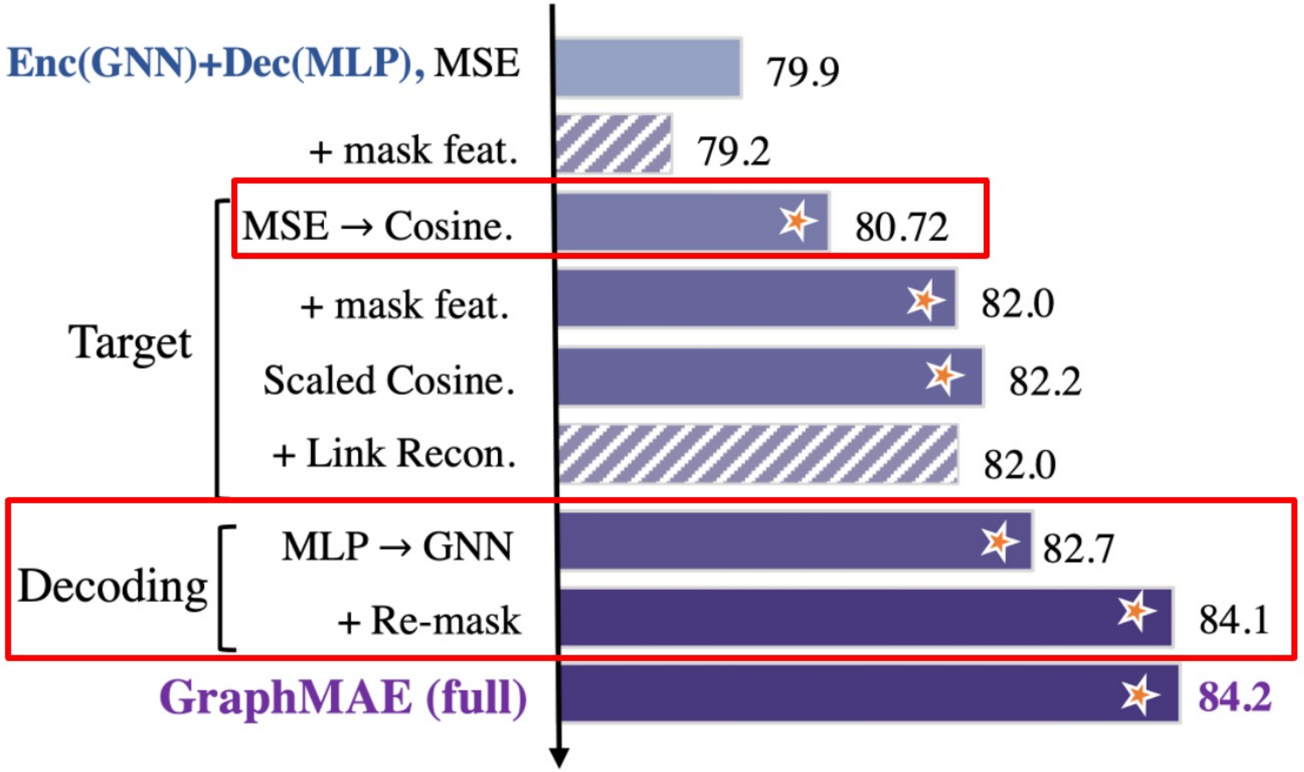
1. Link reconstruction may be over-emphasized.
2. Reconstruction without corruption may not be robust



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

Critical Components

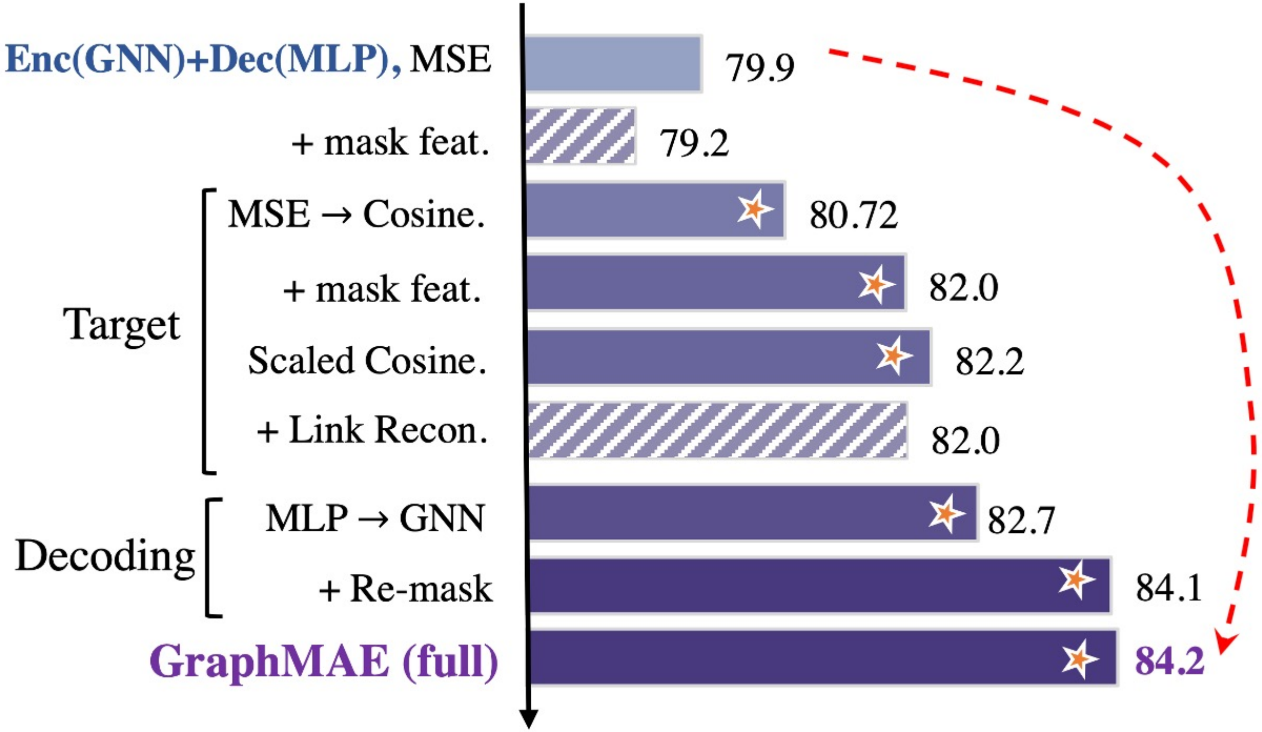
3. Linear/MLP is a less expressive decoding strategy
4. MSE may not be a good criterion for feature reconstruction in graph



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

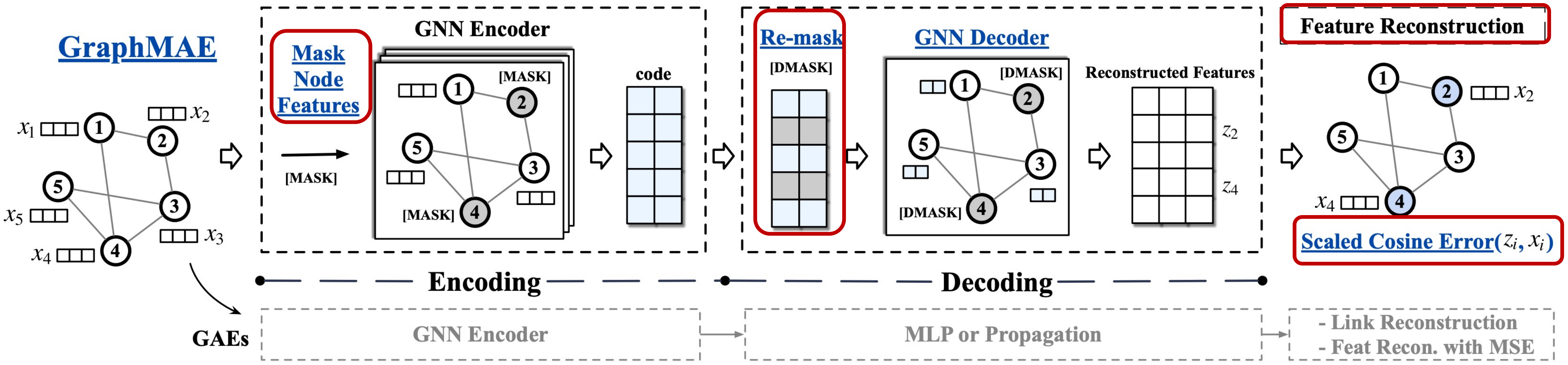
Generative SSL for Graph?

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



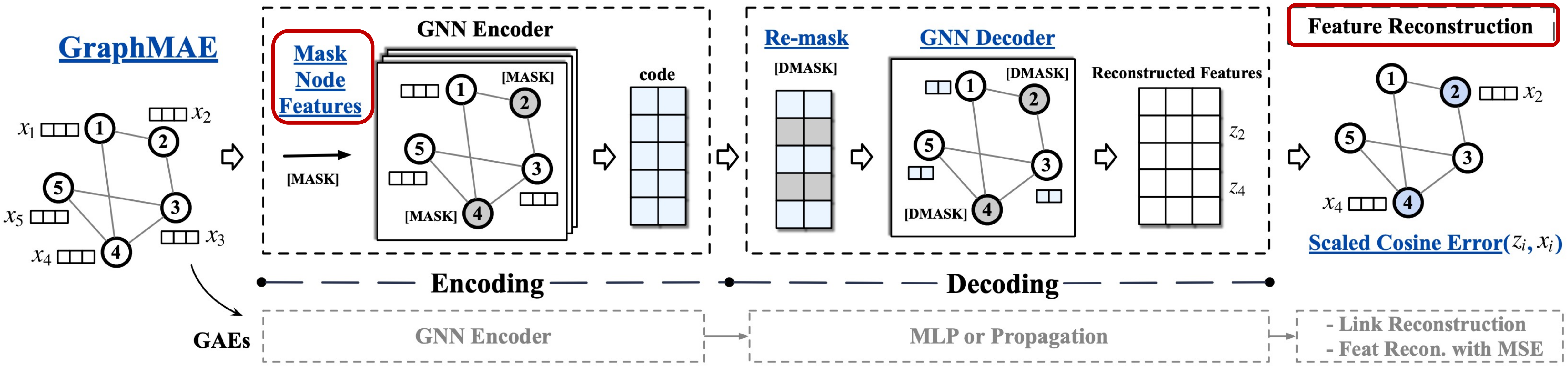
(b) The effect of GraphMAE designs on the performance on Cora dataset.

GraphMAE



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

Masked Feature Reconstruction

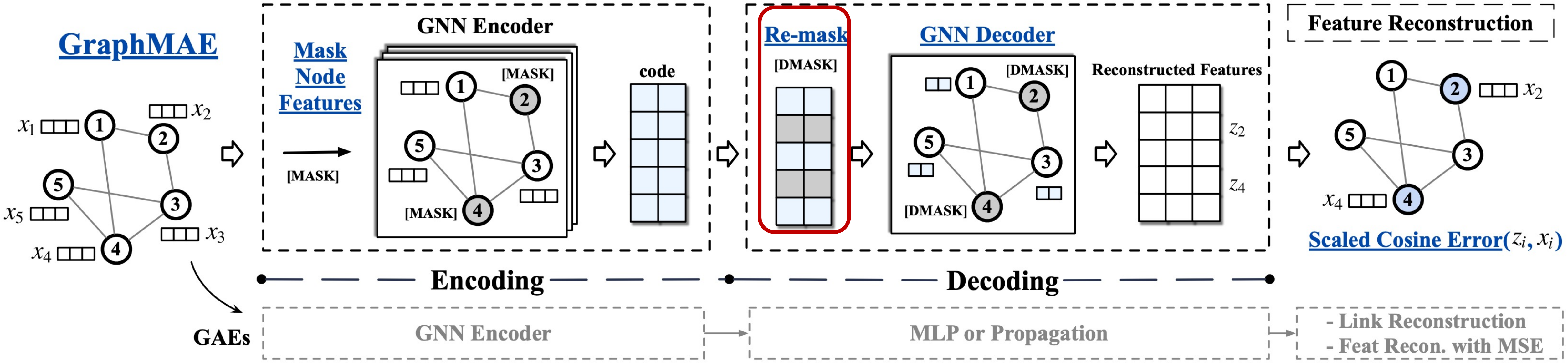


- Feature construction as the learning objective
- Masked feature reconstruction
  - Sample a subset of nodes $\widetilde{V} \subset V$
  - Replace node feature with [MASK]

$$\widetilde{x}_i = \begin{cases} x_{[M]} & v_i \in \widetilde{V} \\ x_i & v_i \notin \widetilde{V} \end{cases}$$

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

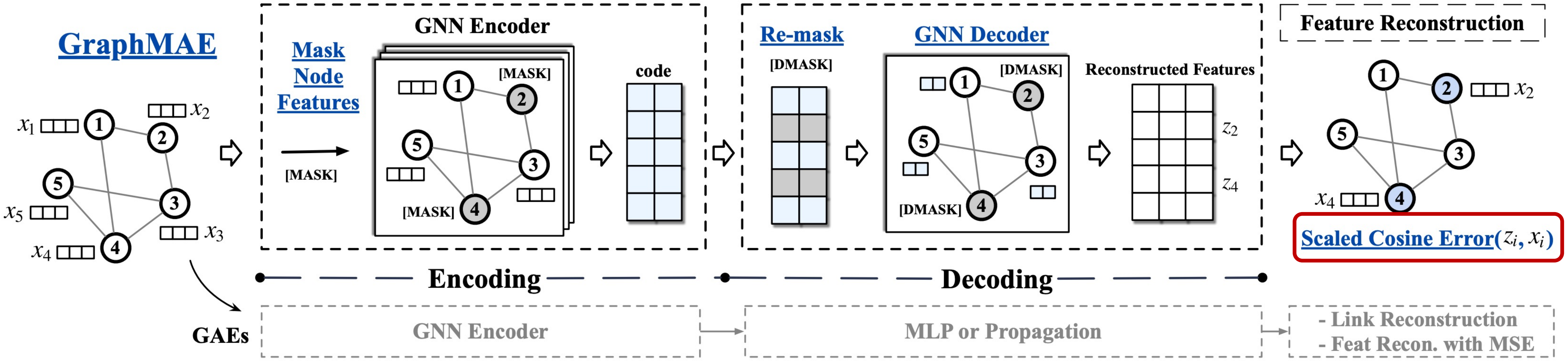
GNNs as Decoder with Re-Mask Decoding



- Use a GNN as the decoder
  - A more expressive decoder helps reconstruct low informative features
- Re-mask node features before decoder
  - Re-mask the “masked” nodes
- $\widetilde{H} = \mathrm{Remask}(H),\ Z = f_D(A, \widetilde{H})$

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

Scaled Cosine Error as the Criterion



- MSE fails, 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

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

Reconstruction Target

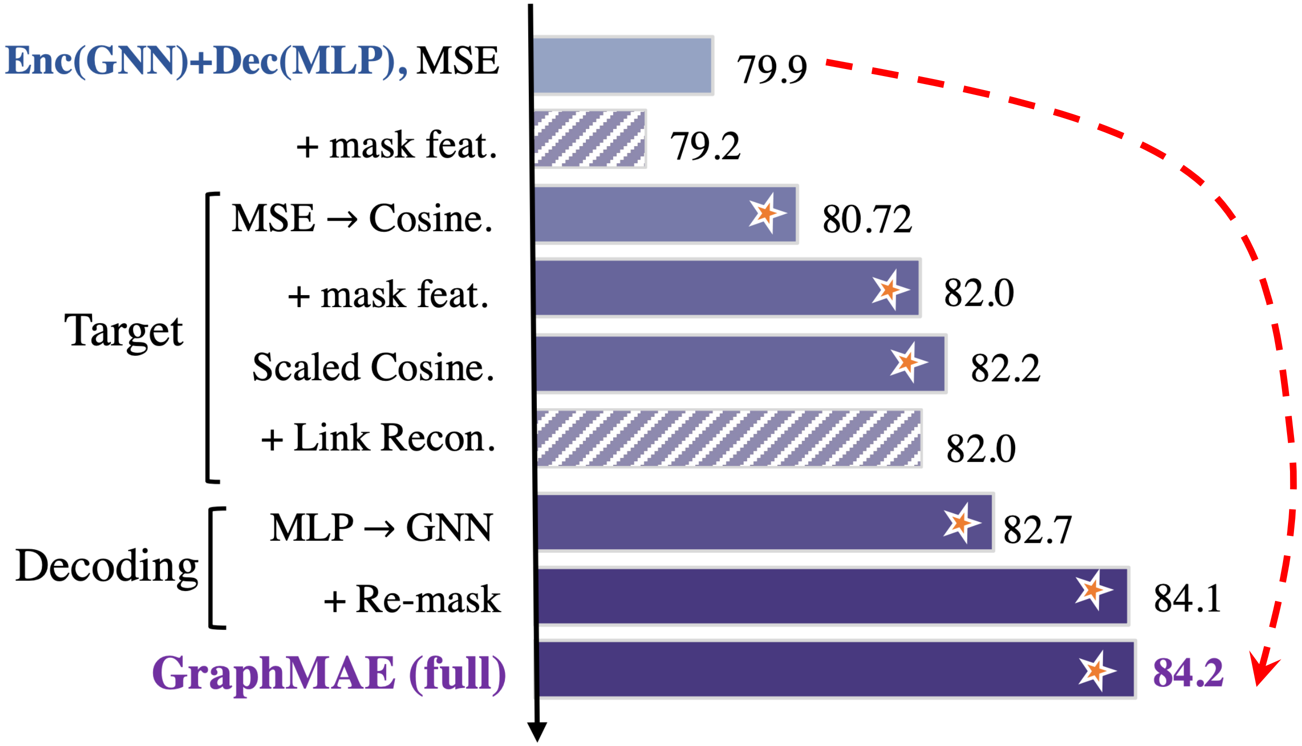
Decoding Strategy

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	✓	✓	✓	✓	✓	$O(N)$

Error Function

Reconstruction Method

GraphMAE



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

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

Downstream Tasks—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.

Contrastive methods

	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	-	-	-
	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

Downstream Tasks—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

Downstream Tasks—Transfer learning

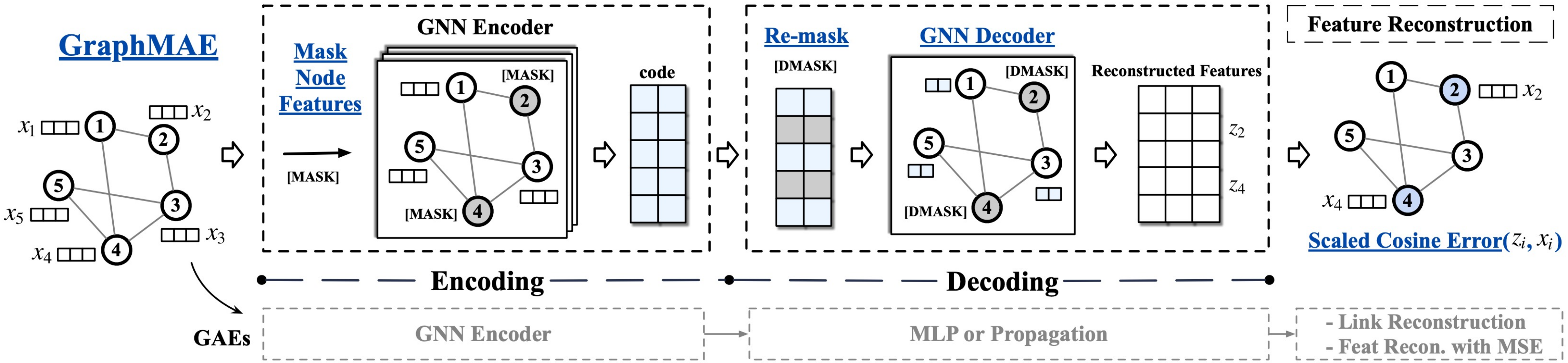
Table 3: Experiment results in transfer learning on molecular property prediction benchmarks. The model is first pre-trained on ZINC15 and then finetuned on the following datasets. We report ROC-AUC(%) scores.

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

Summary of GraphMAE



1. Generative SSL on Graphs vs. Contrastive Learning on Graphs
2. Identify the common issues in current graph autoencoders
3. Present a simple masked graph autoencoder—**GraphMAE**

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

Reflection & Motivation

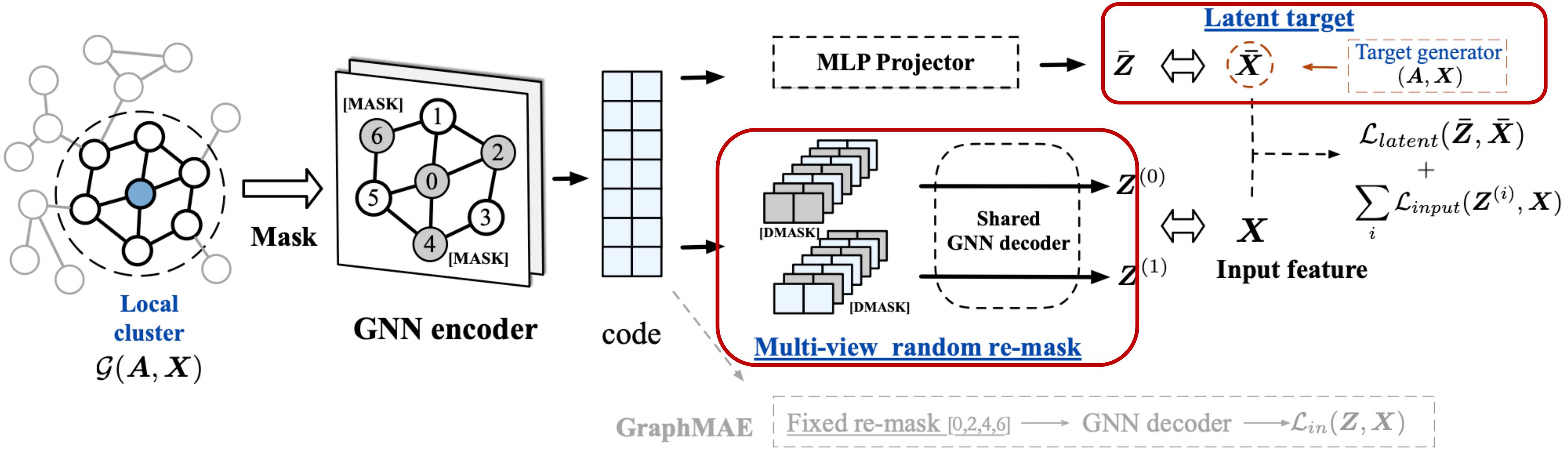
- Problems in masked-feature-prediction
  - – **more *sensitive to the discriminability of input features.***

	Cora raw $\longrightarrow$ *w/ PCA*	PubMed raw $\longrightarrow$ *w/ PCA*
Supervised	83.0 $\longrightarrow$ 82.3 ($\downarrow$ 0.7)	78.0 $\longrightarrow$ 77.0 ($\downarrow$ 1.0)
GraphMAE	84.2 $\longrightarrow$ 82.6 ($\downarrow$ 1.6)	81.1 $\longrightarrow$ 78.9 ($\downarrow$ 2.2)
GraphMAE2	84.5 $\longrightarrow$ 83.5 ($\downarrow$ 1.0)	81.4 $\longrightarrow$ 80.1 ($\downarrow$ 1.3)

- *raw* : the original node features
- *w/ PCA* : the input features are reduced to 50-dimensional vectors using PCA

***Resolution*: imposing regularization on target reconstruction**

The GraphMAE2 Framework



- Multi-view random re-mask decoding
- Latent representation prediction
- Scaling to large-scale graphs with local clustering

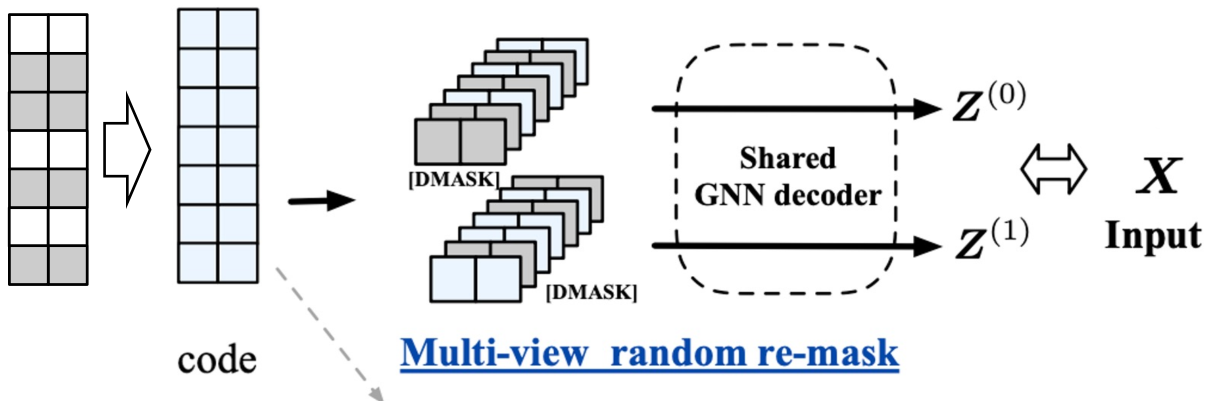
Multi-View Random Re-Mask Decoding

– Avoid representation overfitting to input features

– **Randomly** re-mask representations/code

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

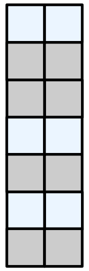
$$\widetilde{\boldsymbol{h}}_i = \begin{cases} \boldsymbol{h}_{[M]} & v_i \in \overline{\mathcal{V}} \\ \boldsymbol{h}_i & v_i \notin \overline{\mathcal{V}} \end{cases}$$



– **Multiple** re-masking

- *K*-different randomly re-masking
- better generalization and effectiveness

$$\mathcal{L}_{input} = \frac{1}{|\widetilde{\mathcal{V}}|} \sum_{j=1}^{K} \sum_{v_i \in \widetilde{\mathcal{V}}} \left(1 - \frac{\boldsymbol{x}_i^{\top} \boldsymbol{z}_i^{(j)}}{\|\boldsymbol{x}_i\| \cdot \|\boldsymbol{z}_i^{(j)}\|}\right)^{\gamma}$$



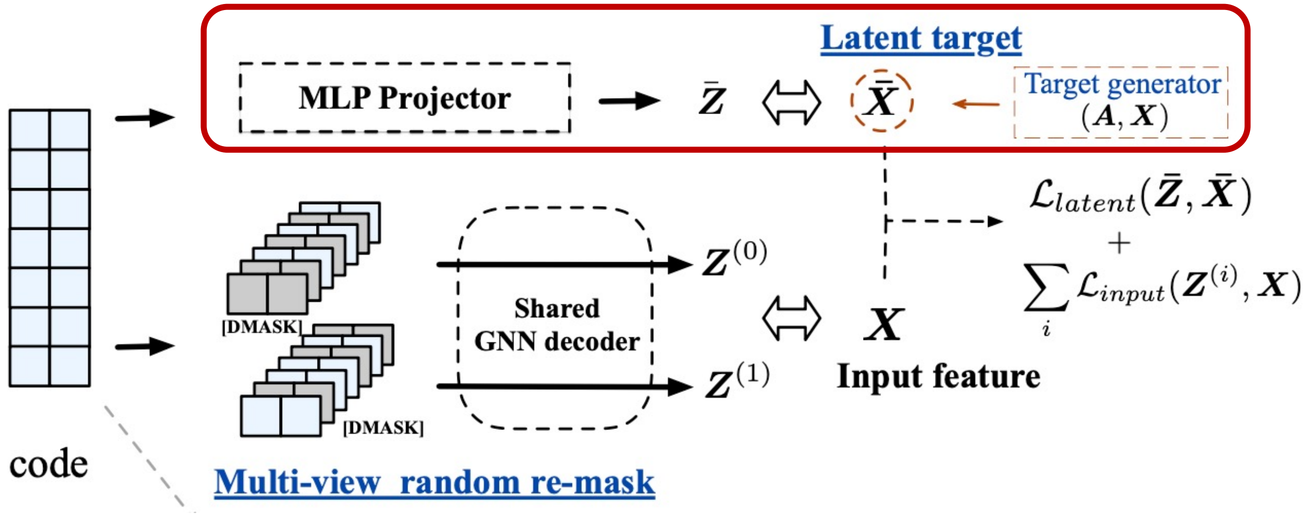
(fixed re-mask in GraphMAE)

Latent Representation Prediction

- Additional informative prediction target
  - Minimally affected by input features & and GNN as a denoiser

- Predicting **masked latent representations**
  - A (*momentum*) target generator $f_{target}(\cdot|\xi)$
  - Prediction: $\bar{Z} = f_E(\text{mask}(G); \theta)$
  - Latent target: $\bar{X} = f_{target}(G; \xi)$
    - $\xi \leftarrow \tau \cdot \xi + (1 - \tau) \cdot \theta$

$$\mathcal{L}_{latent} = \frac{1}{N} \sum_{i}^{N} \left(1 - \frac{\bar{\boldsymbol{z}}_i^\top \bar{\boldsymbol{x}}_i}{\|\bar{\boldsymbol{z}}\| \cdot \|\bar{\boldsymbol{x}}\|}\right)^\gamma$$



Linear Probing

- Setting: training a linear classifier
- GraphMAE2 consistently outperforms all baselines
  - Significantly improves GraphMAE on OGB benchmarks

	Arxiv	Products	MAG	Papers100M
MLP	55.50±0.23	61.06±0.08	39.11±0.21	47.24±0.31
SGC	66.92±0.08	74.87±0.25	54.68±0.23	63.29±0.19
Random-Init	68.14±0.02	74.04±0.06	56.57±0.03	61.55±0.12
CCA-SSG	68.57±0.02	75.27±0.05	51.55±0.03	55.67±0.15
GRACE	69.34±0.01	<u>79.47±0.59</u>	57.39±0.02	61.21±0.12
BGRL	70.51±0.03	78.59±0.02	57.57±0.01	62.18±0.15
GGD[1]	-	75.70±0.40	-	<u>63.50±0.50</u>
GraphMAE	<u>71.03±0.02</u>	78.89±0.01	<u>58.75±0.03</u>	62.54±0.09
GraphMAE2	**71.89**±0.03	**81.59**±0.02	**59.24**±0.01	**64.89**±0.04

Code: https://github.com/THUDM/GraphMAE2 OGB benchmarks

Ablation studies

Component Ablation of GraphMAE2

- Decoding strategies both bring benefits
- GraphMAE2 surpasses all baselines with the same sampling strategy
  - Using local clusters brings further improvement

Table 6: Ablation studies of GraphMAE2 key components.

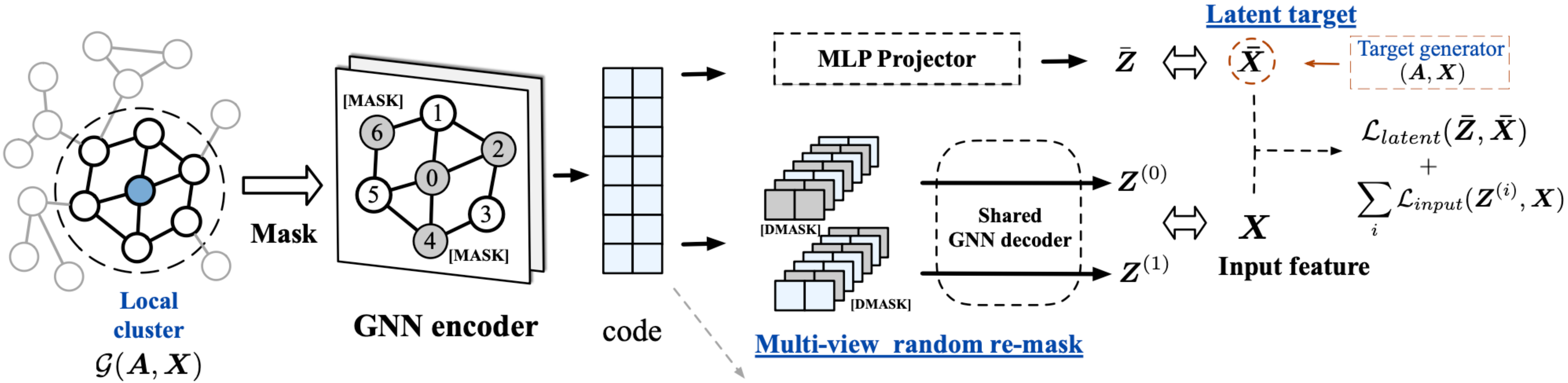
	Products	MAG	Papers100M
GraphMAE2	81.59±0.02	59.24±0.01	64.89±0.04
w/o random remask	81.04±0.03	59.01±0.02	64.16±0.02
w/o latent rep pred.	80.01±0.02	58.87±0.02	62.98±0.01
w/o input recon.	76.88±0.02	55.20±0.02	59.20±0.00
GraphMAE	78.89±0.01	58.75±0.03	62.54±0.09

Table 7: Ablation study on sampling strategy. "SAINT" refers to GraphSAINT, "Cluster" refers to Cluster-GCN, and "LC" refers the presented local clustering algorithm.

	Strategy	Products	MAG	Papers100M
GRACE	*SAINT*	79.47±0.59	57.39±0.02	61.21±0.12
BGRL	*SAINT*	78.59±0.02	57.57±0.01	62.18±0.15
GraphMAE2	*SAINT*	80.96±0.03	58.75±0.03	64.21±0.11
GraphMAE2	*Cluster*	79.35±0.05	58.05±0.02	63.77±0.11
GraphMAE2	*LC*	81.59±0.02	59.24±0.01	64.89±0.12

Code: https://github.com/THUDM/GraphMAE2

GraphMAE2 Summary



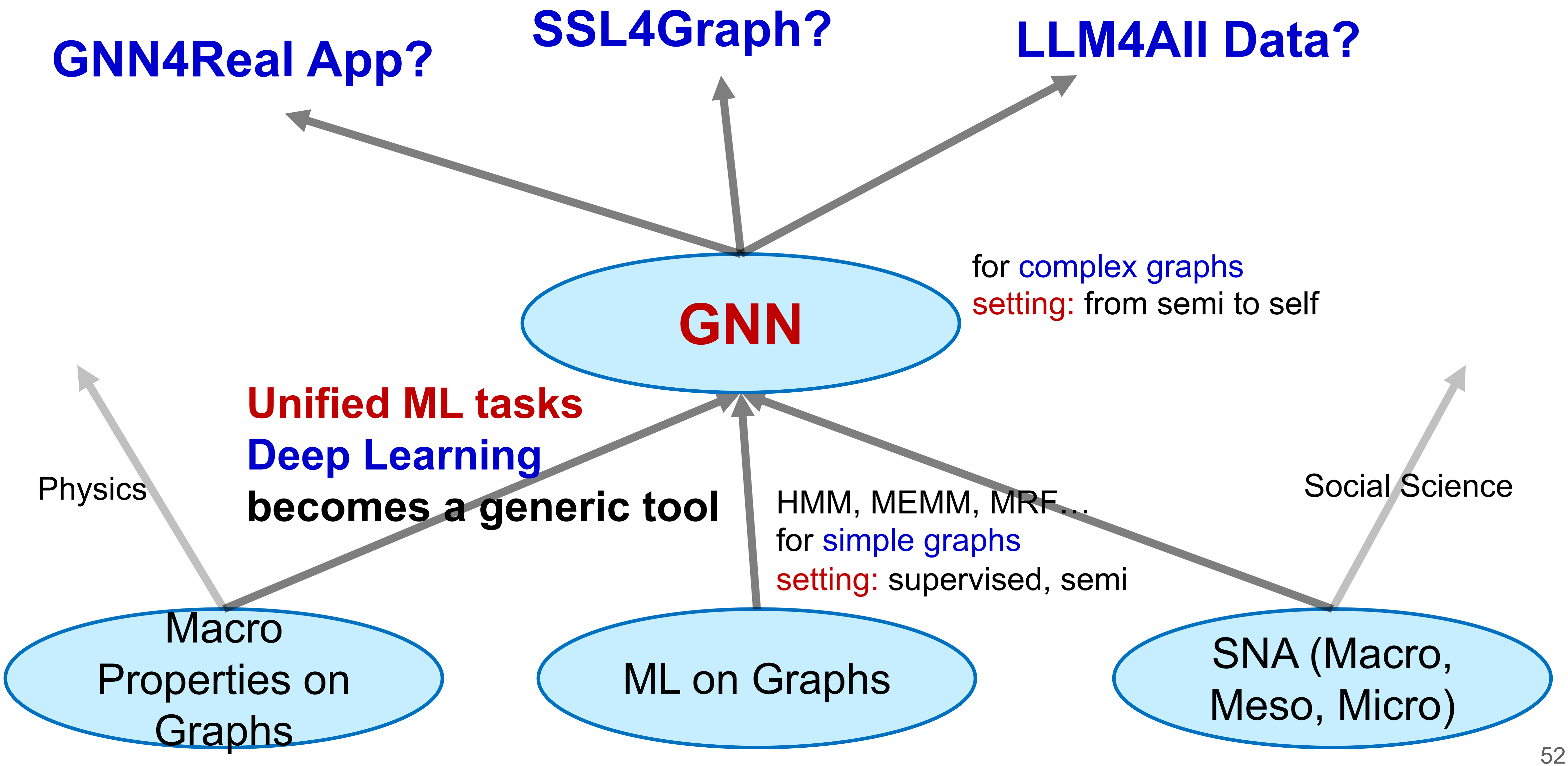
- Analyze the problem in masked feature prediction
- Present GraphMAE2 with improved decoding strategies
- GraphMAE2 achieves promising performance in large-scale graphs

GraphMAE2: https://github.com/THUDM/GraphMAE2
GraphMAE: https://github.com/THUDM/GraphMAE

- Now we have both contrastive and generative models for Graphs

- What is the future?

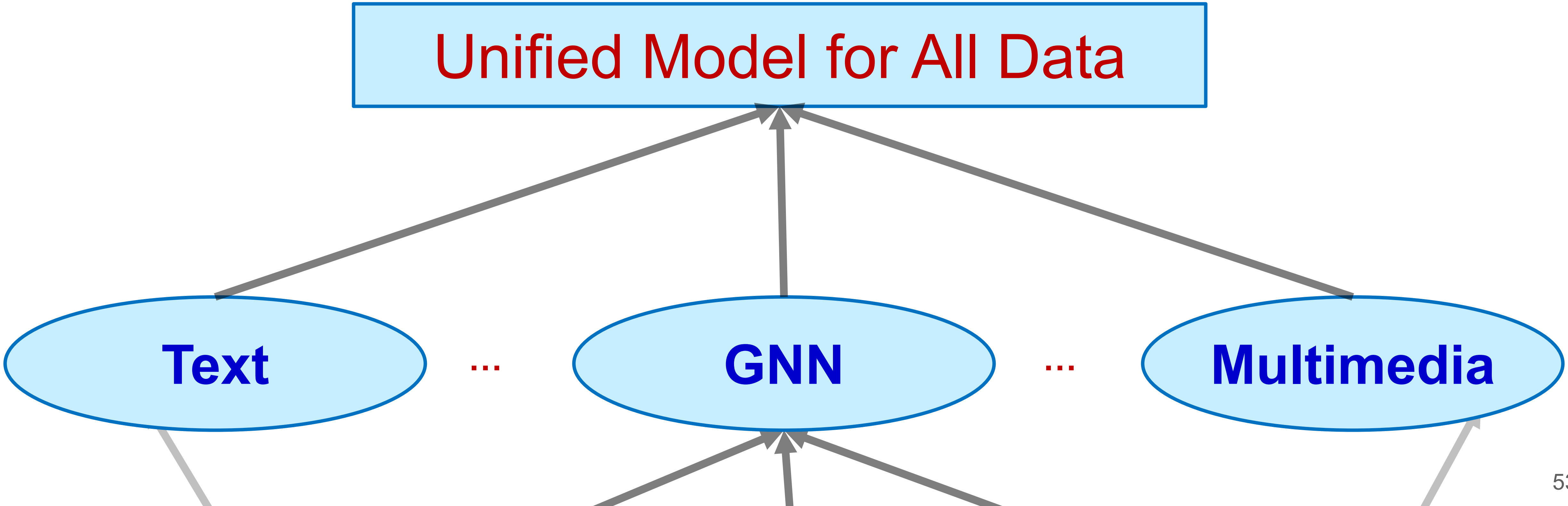
What is the direction of GNN?



What is the direction of GNN?

for complex graphs, features, multimodal
setting: self

for real applications, science
setting: fine-tune



Thank you !

ChatGLM-6B Public

ChatGLM-6B: An Open Bilingual Dialogue Language Model | 开源双语对话语言模型

Python ☆ 21.9k 2.6k

GLM-130B Public

GLM-130B: An Open Bilingual Pre-Trained Model (ICLR 2023)

Python ☆ 5.1k 365

CodeGeeX Public

CodeGeeX: An Open Multilingual Code Generation Model

Python ☆ 4.8k 316

CogView Public

Text-to-Image generation. The repo for NeurIPS 2021 paper "CogView: Mastering Text-to-Image Generation via Transformers".

Python ☆ 1.4k 163

CogVideo Public

Text-to-video generation. The repo for ICLR2023 paper "CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers"

Python ☆ 2.8k 286

cogdl Public

CogDL: A Comprehensive Library for Graph Deep Learning (WWW 2023)

Python ☆ 1.4k 300

slides

https://github.com/THUDM

@thukeg