南京航空航天大學

Nanjing University of Aeronautics and Astronautics

模式分析与机器智能
工业和信息化部重点实验室

MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training

Jiezhong Qiu
qiujz16@mails.tsinghua.edu.cn
Tsinghua University

Qibin Chen
cqb19@mails.tsinghua.edu.cn
Tsinghua University

Yuxiao Dong
yuxdong@microsoft.com
Microsoft Research, Redmond

Jing Zhang
zhang-jing@ruc.edu.cn
Remin University

Hongxia Yang
yang.yhx@alibaba-inc.com
DAMO Academy, Alibaba Group

Ming Ding
dm18@mails.tsinghua.edu.cn
Tsinghua University

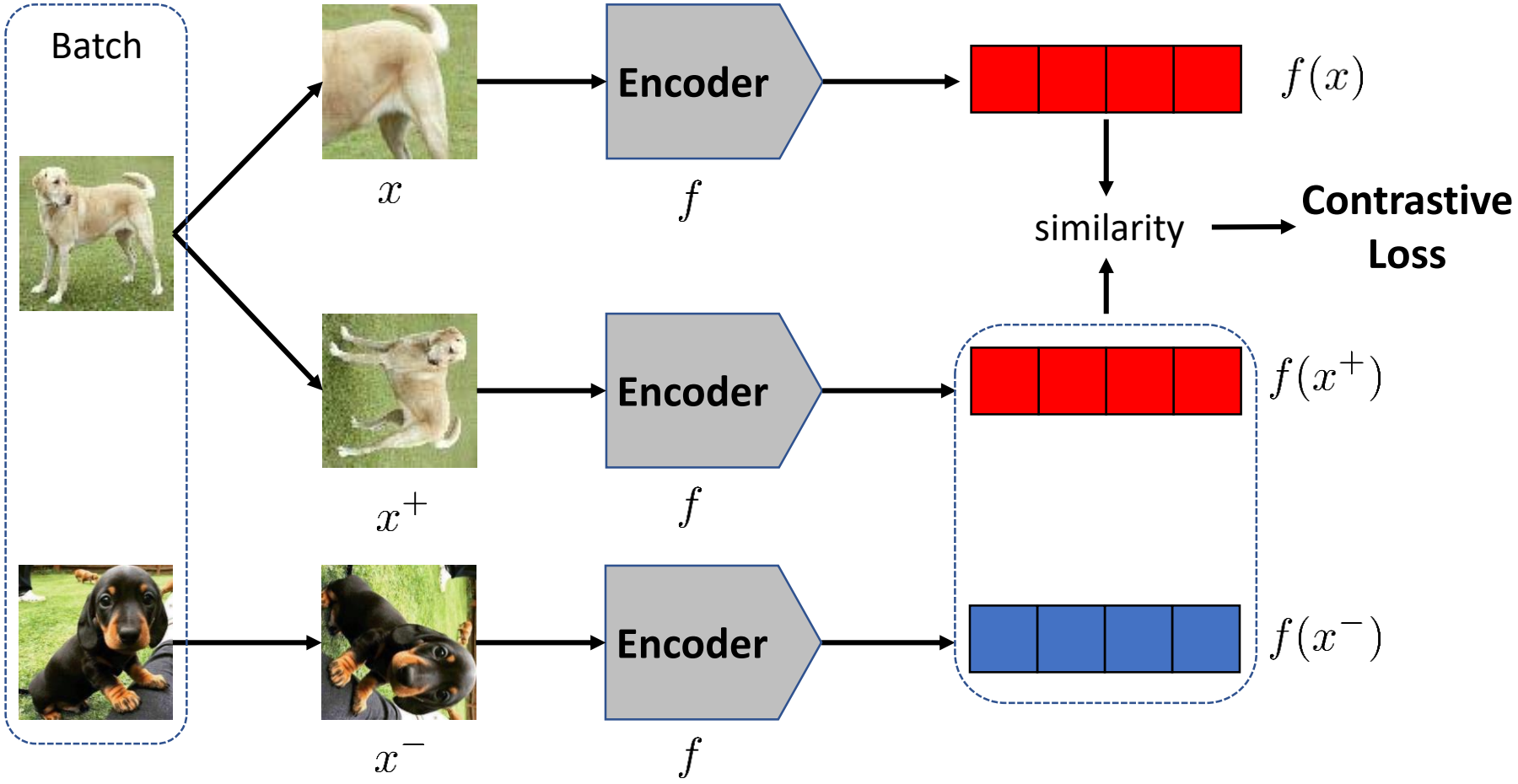
Kuansan Wang
kuansan.wang@microsoft.com
Microsoft Research, Redmond

Jie Tang*
jietang@tsinghua.edu.cn
Tsinghua University

KDD 2020

Background

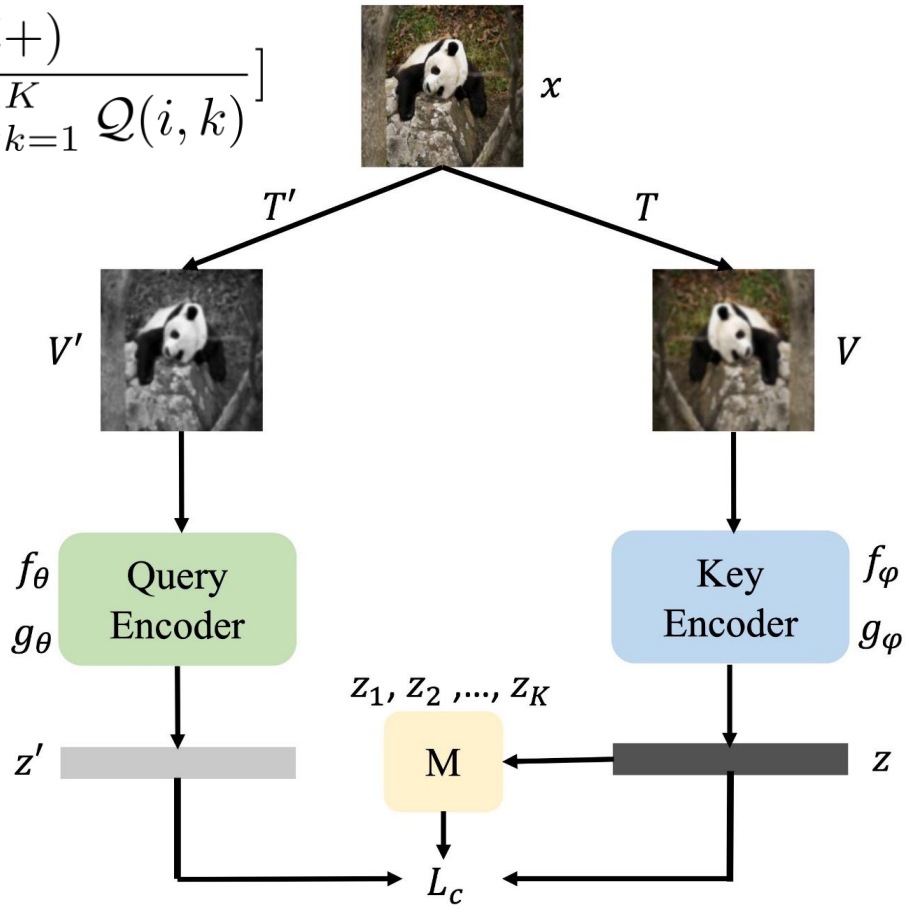
Contrastive Learning



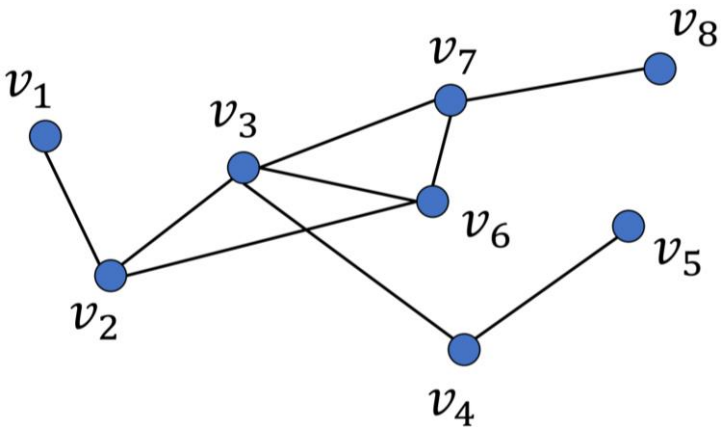
$$\text{sim}\left(f(x), f(x^+)\right) \gg \text{sim}\left(f(x), f(x^-)\right)$$

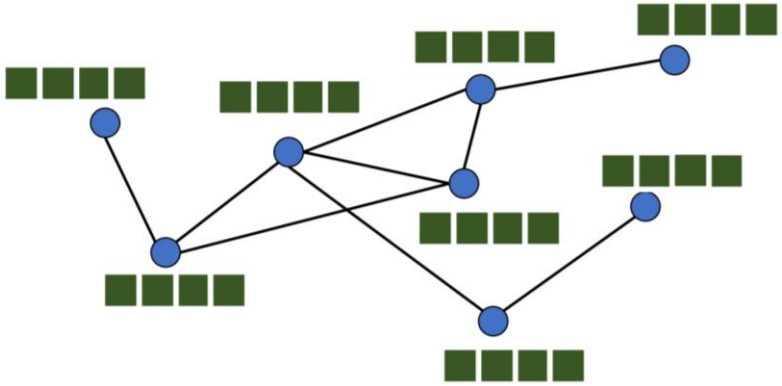
Contrastive Learning

- Contrastive Loss: **InfoNCE**
  - $\mathcal{L}_C = \mathbb{E}_{i \in B}[-\log \frac{\mathcal{Q}(i, i+)}{\mathcal{Q}(i, i+) + \sum_{k=1}^{K} \mathcal{Q}(i, k)}]$
- Similarities of Positive Pairs
  - $\mathcal{Q}(i, i+) = \exp\left(sim(z_i', z_i)/\tau\right)$
- Similarities of Negative Pairs
  - $\mathcal{Q}(i, k) = \exp\left(sim(z_i', z_k)/\tau\right)$
- Cosine similarity
  - $sim(z_i', z_k) = \frac{{z_i'}^T z_k}{\|z_i'\| \cdot \|z_k\|}$



Graph Neural Network





$$\mathcal{V} = \{v_1, \ldots, v_N\}$$

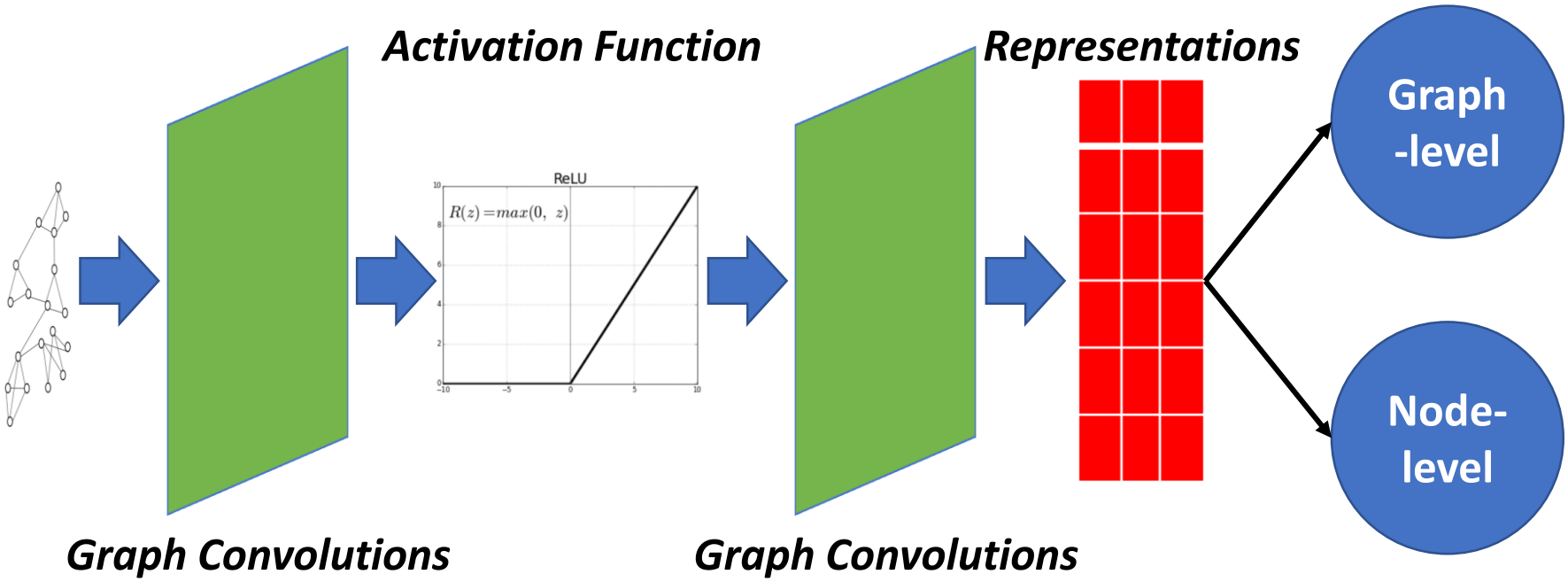
$$\mathcal{E} = \{e_1, \ldots, e_M\}$$

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

Feature matrix: $\mathbf{X} \in \mathbb{R}^{N \times d}$

Adjacency matrix: $\mathbf{A} \in \mathbb{R}^{N \times N}$

Graph Neural Network



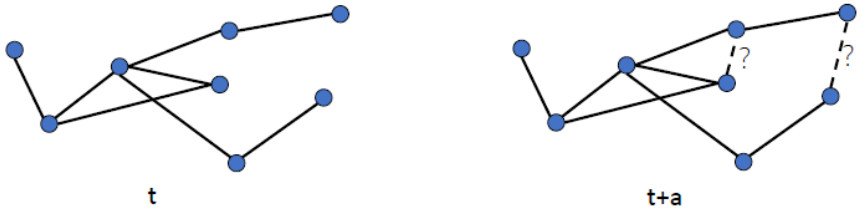
$$\mathbf{H}^{l+1} = h(\hat{\mathbf{A}}\mathbf{H}^{l}\mathbf{W}^{l})$$

- $\mathbf{W}^{l} \in \mathbb{R}^{d \times d'}$ is a transformation matrix to be learned
- $\hat{\mathbf{A}} \in \mathbb{R}^{n \times n}$ is the normalized version of correlation matrix $\mathbf{A}$
- $h(\cdot)$ denotes a non-linear operation

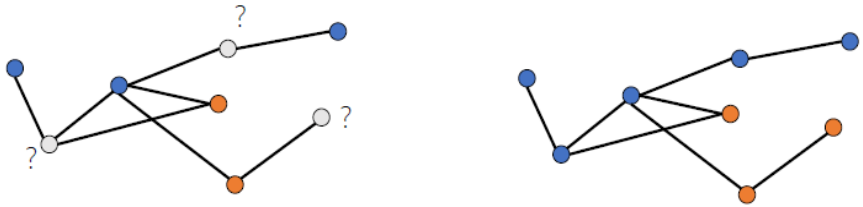
Graph Neural Network

Tasks on Graph-Structured Data

Node-level

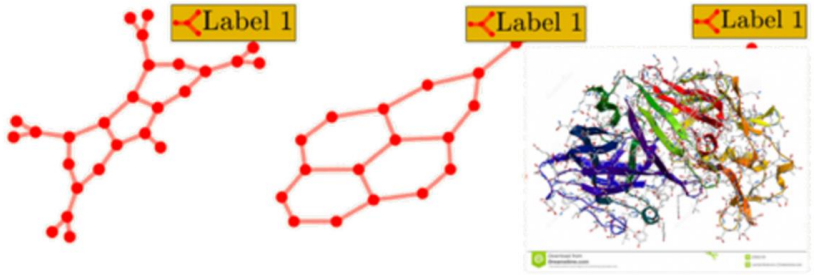


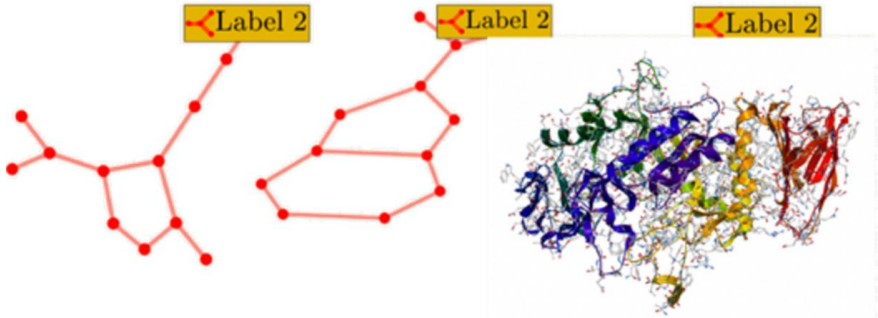
Link Prediction



Node Classification

Graph-level

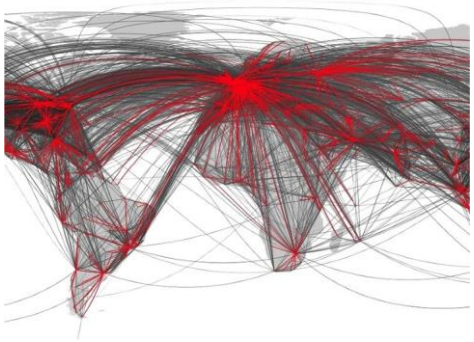


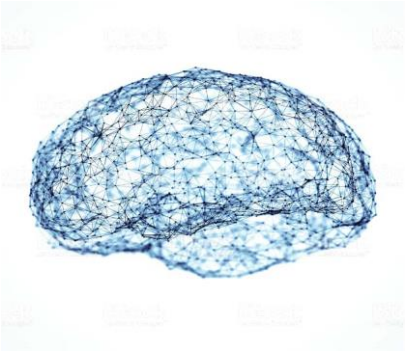


Graph Classification

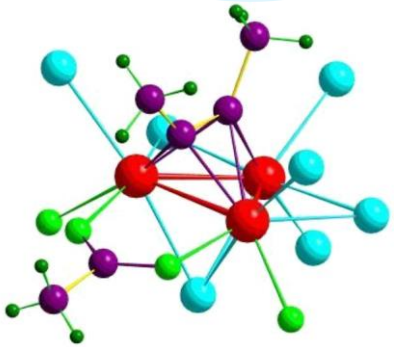
Motivation

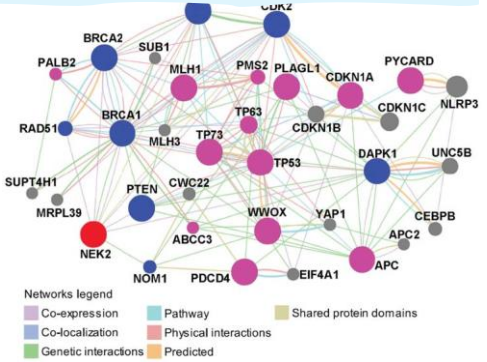
Real World Graph





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



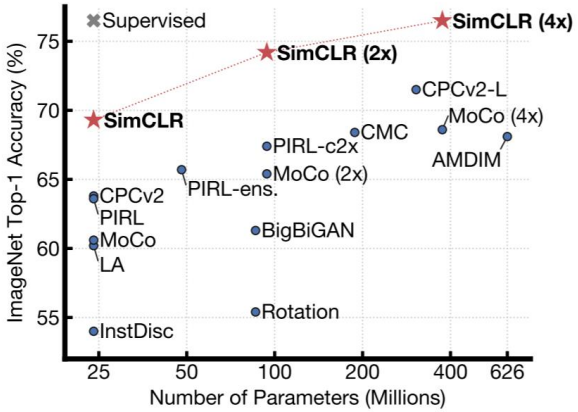


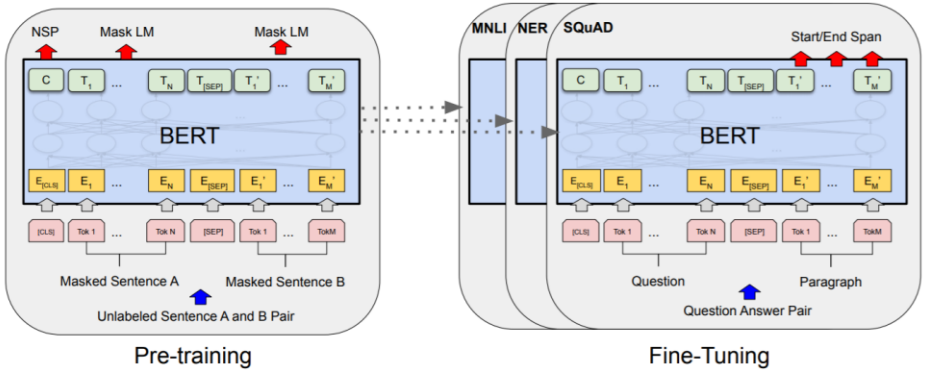
Web Graphs

Molecular Graphs

Gene Graphs

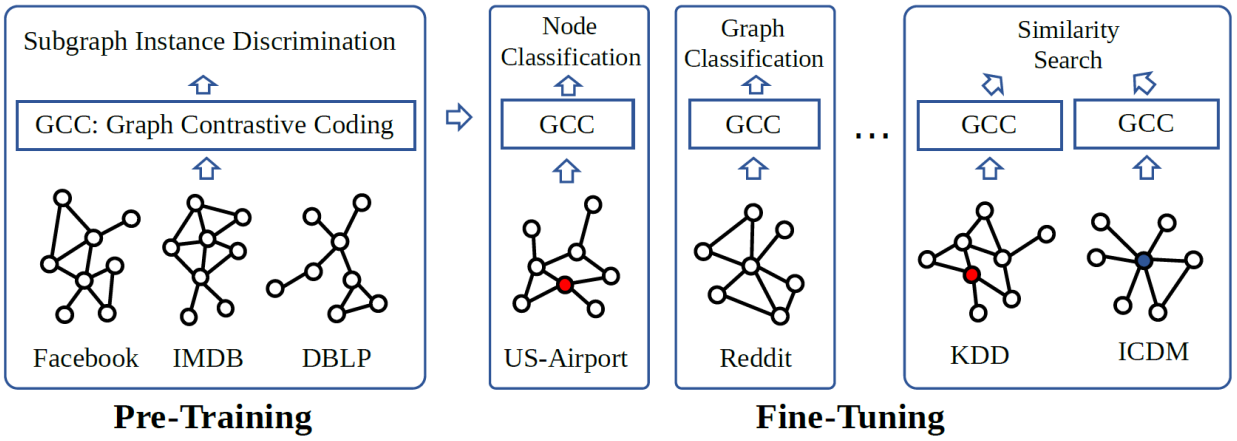
Pre-training and Fine-tuning





Computer Vision
ResNet
ImageNet

Natural Language Processing
BERT
Wikipedia + Book corpus

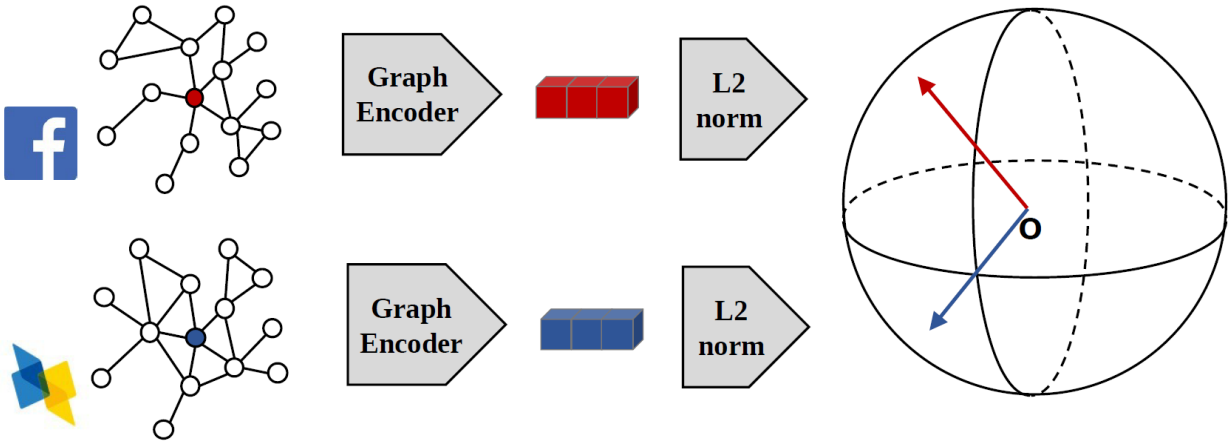


Graph Learning
GCC

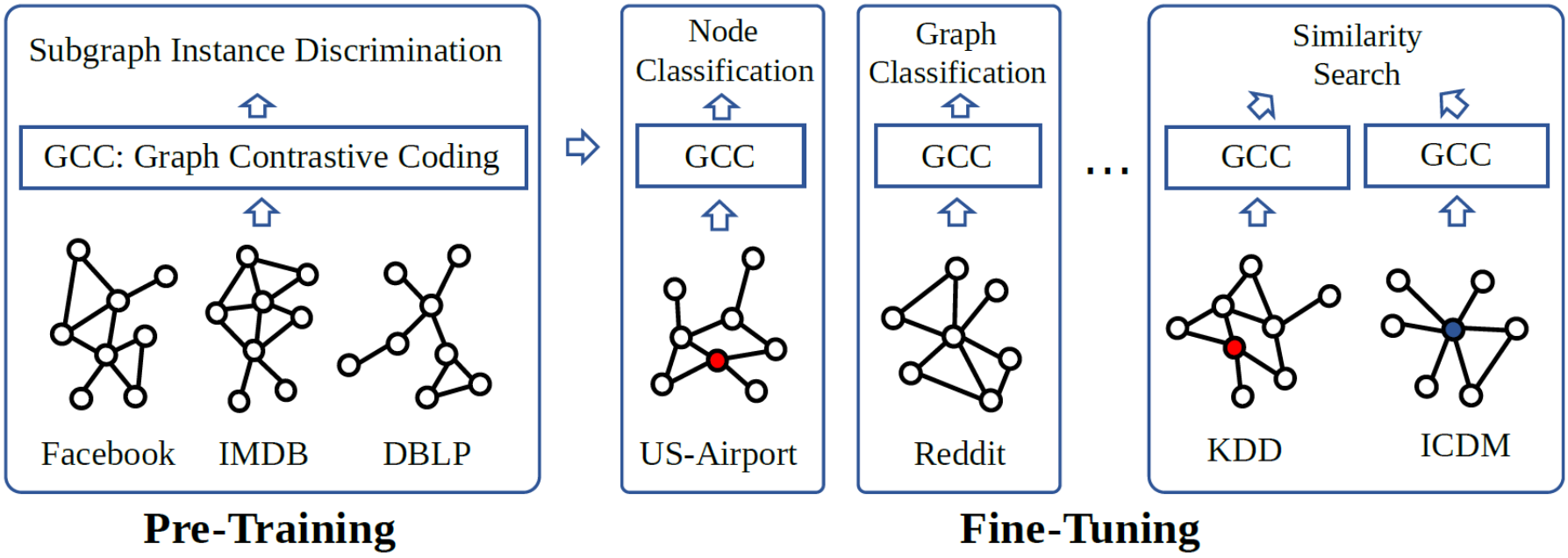
Framework

The GNN Pre-training

- 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 graph from various sources, even unseen during training time.



Graph Contrastive Coding (GCC)



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

GCC Pre-training

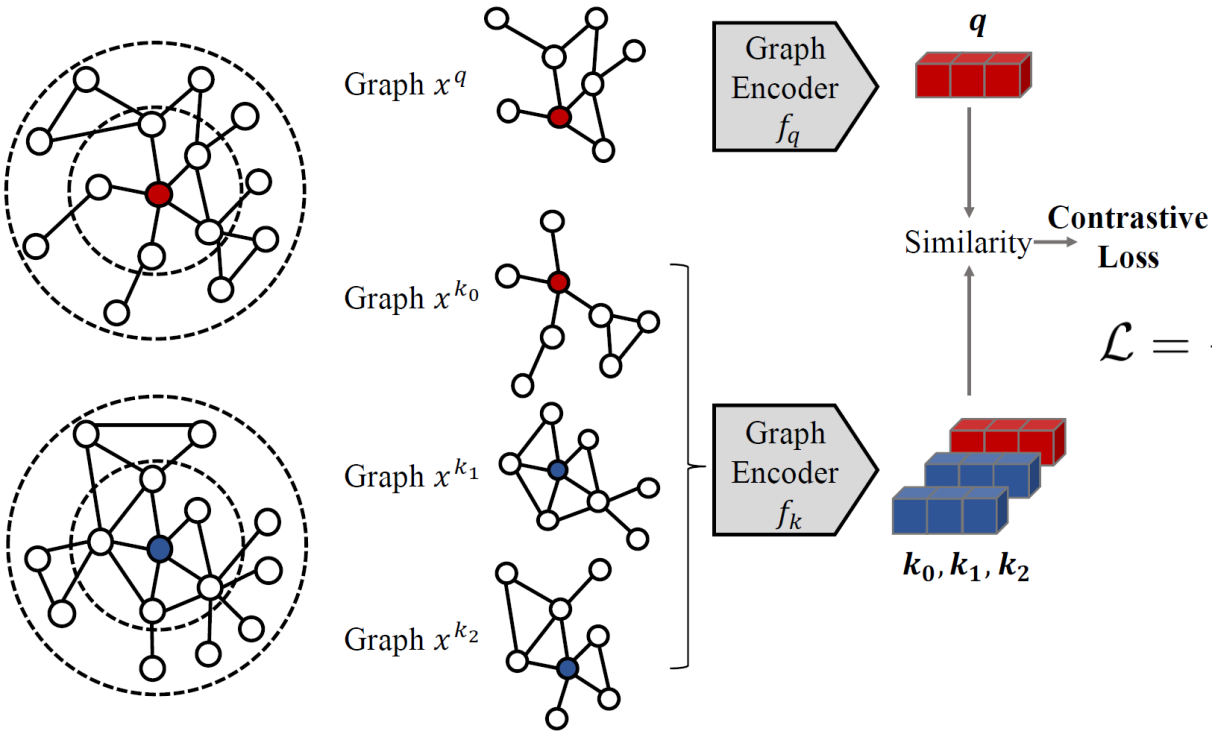
- **Pre-training Task:** Instance Discrimination.
- **InfoNCE Loss:** output instance representations that are capable of capturing the similarities between instances.

$$\mathcal{L}_C = \mathbb{E}_{i \in B}[-\log \frac{\mathcal{Q}(i, i+)}{\mathcal{Q}(i, i+) + \sum_{k=1}^{K} \mathcal{Q}(i, k)}]$$

- *Contrastive learning for graphs ?*
  - Questions 1: *How to define instances in graphs?*
  - Questions 2: *How to define (dis)similar pairs?*
  - Questions 3: *How to encode the instances* ?

GCC Pre-training

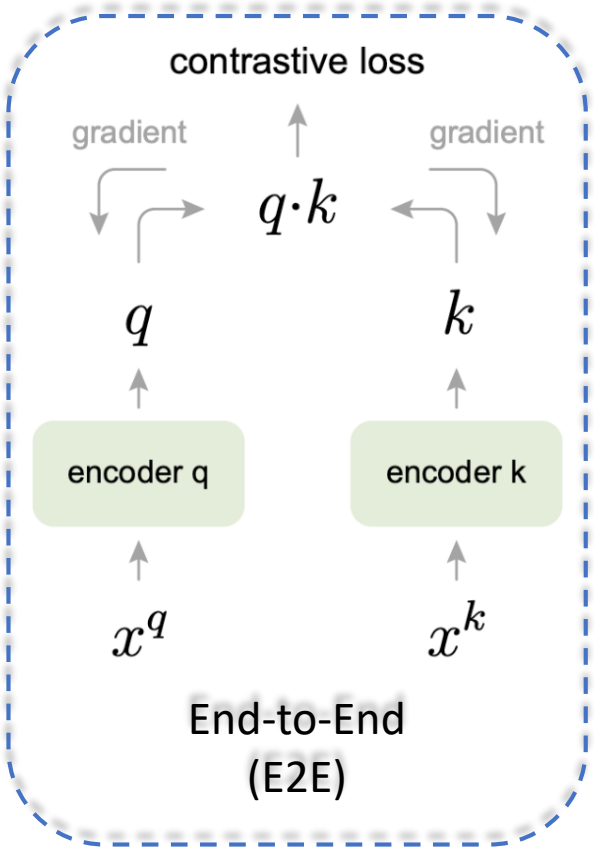
- *Contrastive learning for graphs ?*
  - Questions 1: *How to define instances in graphs?*
  - Questions 2: *How to define (dis)similar pairs?*
  - Questions 3: *How to encode the instances* ?

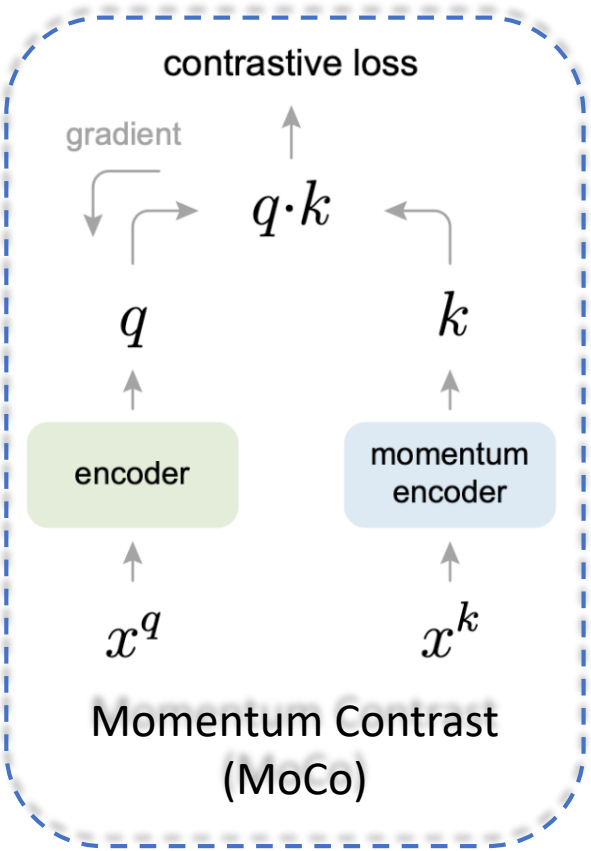


$$\mathcal{L} = -\log \frac{\exp(\mathbf{q}^T \mathbf{k}_+ / \tau)}{\sum_{i=0}^{K} \exp(\mathbf{q}^T \mathbf{k}_i / \tau)}$$

GCC Pre-training: Learning Algorithms

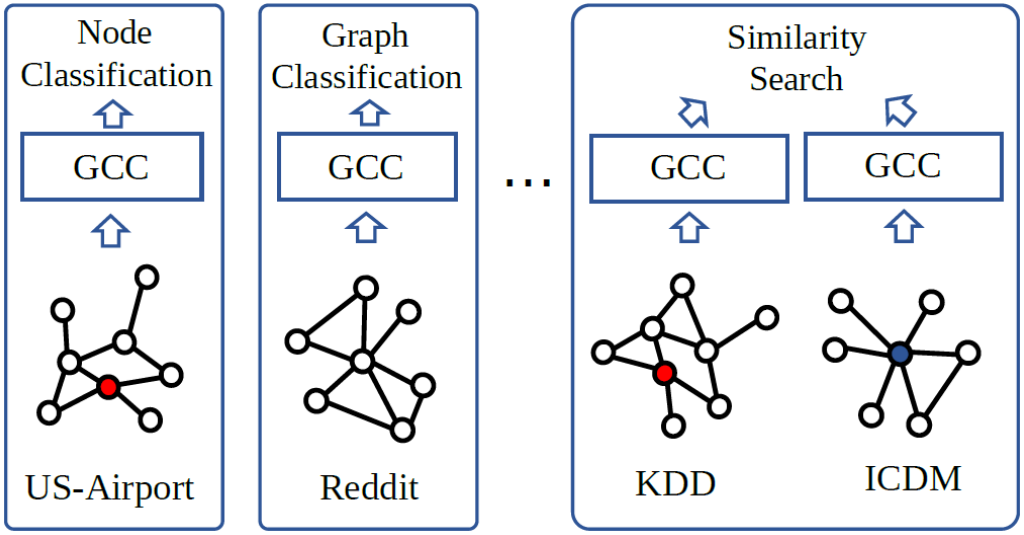
- Optimizing Contrastive Loss
  - Encoded query $\boldsymbol{q}$
  - $\boldsymbol{K+1}$ encoded keys $\{\boldsymbol{k_0}, \cdots, \boldsymbol{k_K}\}$



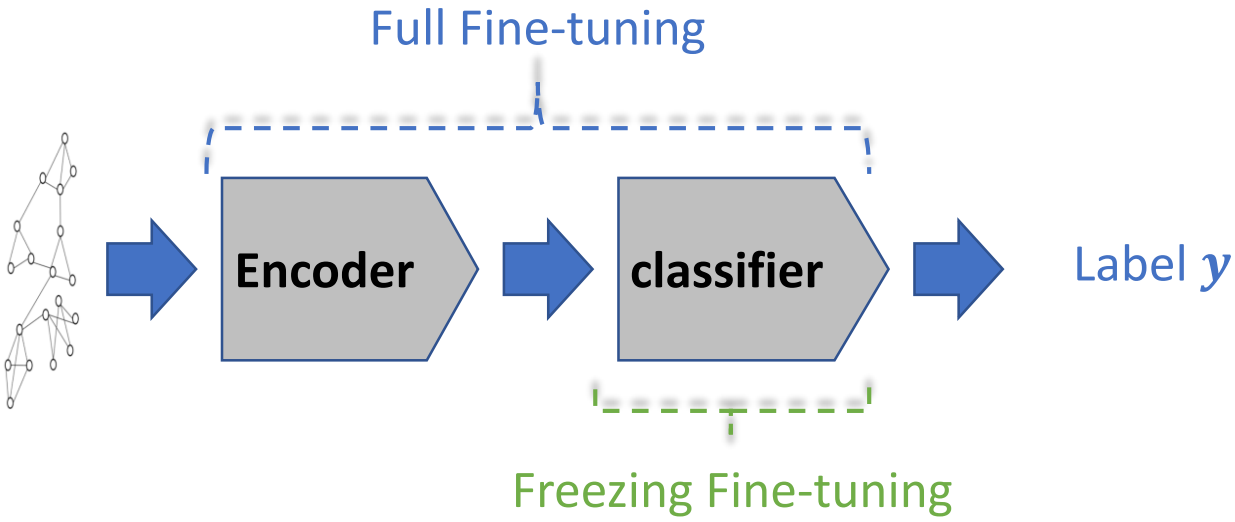


GCC Fine-tuning

Down-stream Tasks



Fine-tuning



Experiment

GCC Pre-training / Fine-tuning

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

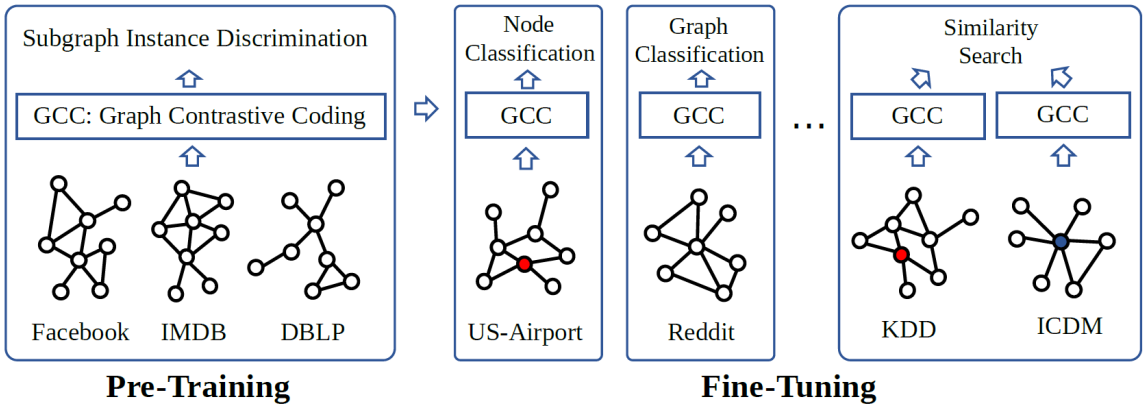
Table 1: Datasets for pre-training, sorted by number of vertices.

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

Academic graphs

Social graphs

- *Three* fine-tuning tasks:
  - Node classification
  - Graph classification
  - Top-k similarity search



Task 1: Node Classification

- **Node classification**: predict unknown node labels in a partially labeled network.

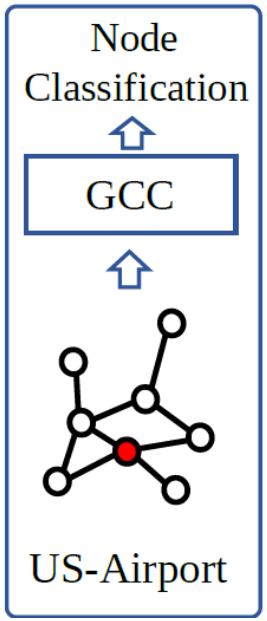


Table 2: Node classification.

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

- **Graph classification**: predict the label of the graph.

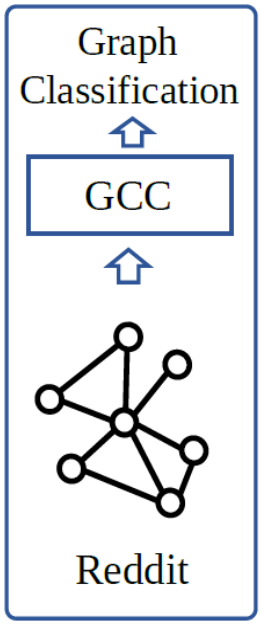


Table 3: Graph classification.

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

- **Datasets**: the conference co-author graphs of KDD, ICDM, SIGIR, CIKM, SIGMOD and ICDE.
- Top-k similarity search: find the most similar vertex v from G_1 for each vertex u in G_2.

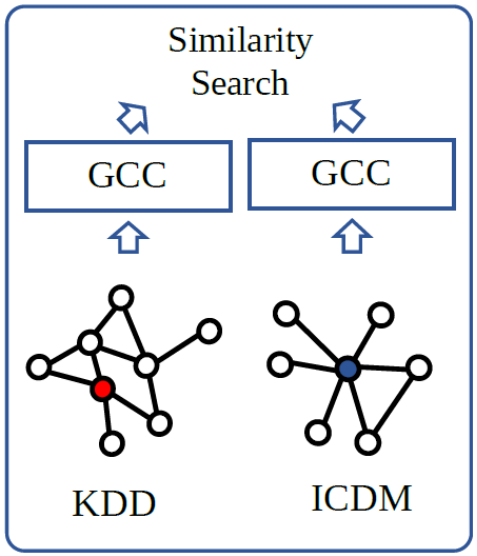


Table 4: Top-k similarity search ($k = 20, 40$).

	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

Task 3: Top-k similarity search

- **Datasets**: the conference co-author graphs of KDD, ICDM, SIGIR, CIKM, SIGMOD and ICDE.
- Top-k similarity search: find the most similar vertex v from G_1 for each vertex u in G_2.

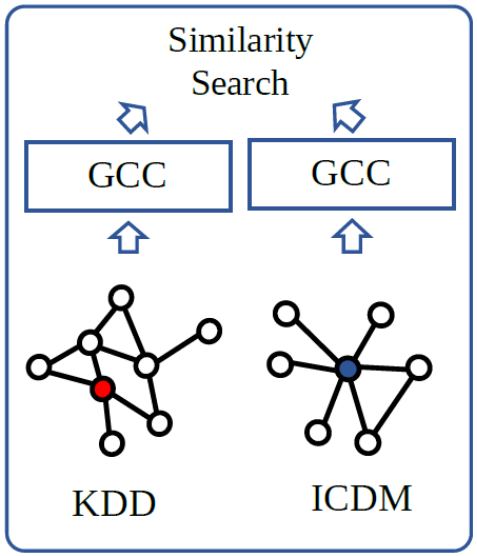
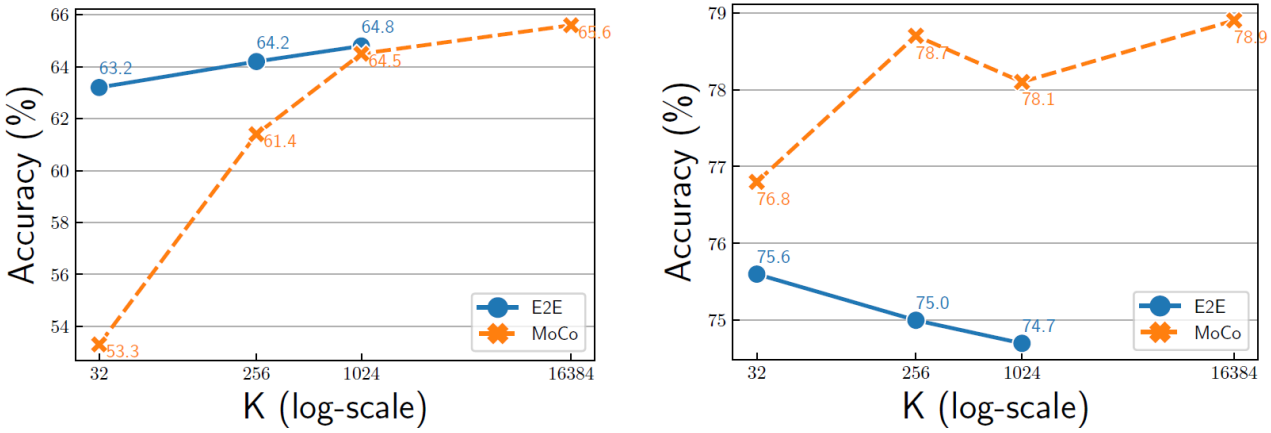


Table 4: Top-k similarity search ($k = 20, 40$).

	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

Ablation study: contrastive loss mechanisms



(a) US-Airport (b) COLLAB

Figure 4: Comparison of contrastive loss mechanisms.

The effect of a large dictionary size is not as significant as reported in computer vision tasks.


Thanks!