Graph Self-Supervised Learning: Taxonomy, Frontiers, and Applications

Yixin Liu[1], Yizheng Zheng[1], Ming Jin[1], Feng Xia[2], Shirui Pan[3]

[1] Monash University

[2] RMIT University

[3] Griffith University

June 18 1:30pm-3:30pm, 2023 (GMT +10)

Tutorial outline

Time	Content	Part	Presenter
20 min	• **Introduction and background** • Graph analytics and graph neural networks • Background of graph self-supervised learning	Part 1	**Shirui Pan**
30 min	• **Taxonomy of graph self-supervised learning** • Uniform framework • Categories of GSSL • Representative methods	Part 2	**Ming Jin**
30 min	• **Frontiers of graph self-supervised learning** • graph self-supervised learning • Efficient graph self-supervised learning • Automatic graph self-supervised learning	Part 3	**Yizhen Zheng**
30 min	• **Applications of graph self-supervised learning** • Recommender system • Outlier detection • More applications	Part 4	**Yixin Liu**
10 min	• **Future directions and conclusion** • Potential directions of graph self-supervised learning • Conclusion	Part 5	**Yixin Liu**

Part 1:Introduction and background

- Graph analytics
- Graph neural networks
- Graph self-supervised learning: Background

What is graphs?

Example: A Social Network Graph

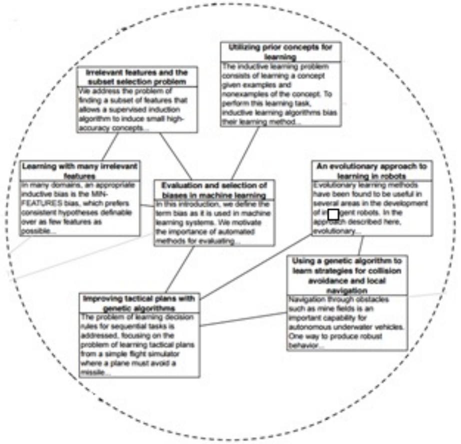
A Graph has nodes/vertices and edges.

Nodes/vertices → a person in the social network

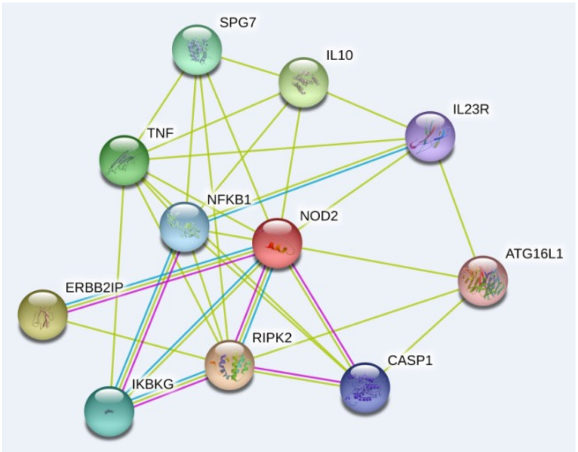
Edges → Connection between people

Graphs in real-world applications

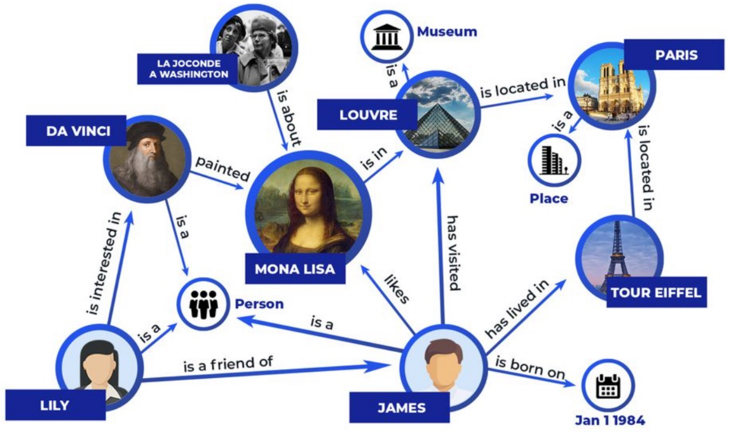
Social Networks



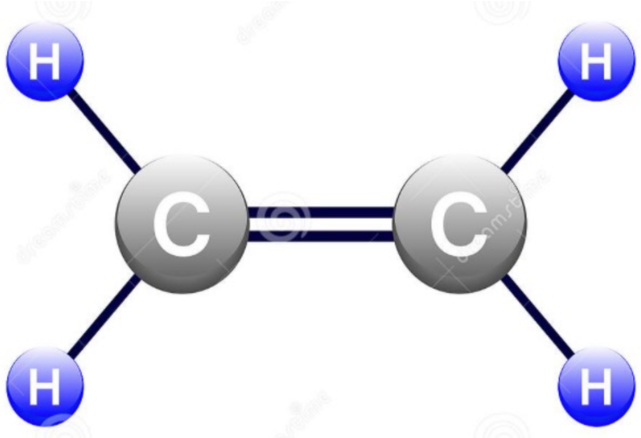
Bibliography Networks



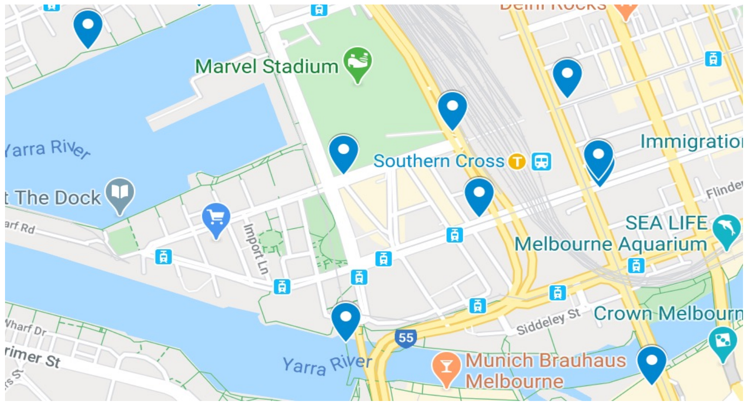
Protein Interaction Networks



Knowledge Graphs

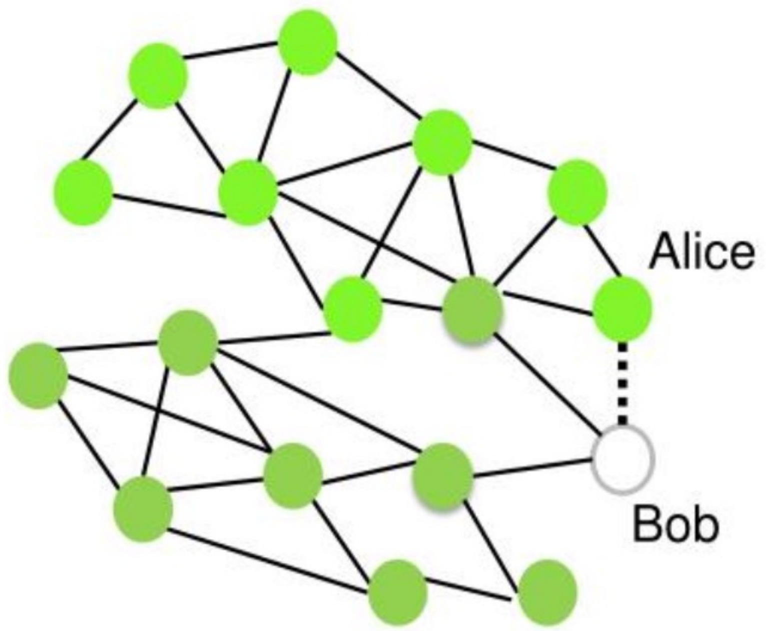


Chemical Compounds



Traffic Networks

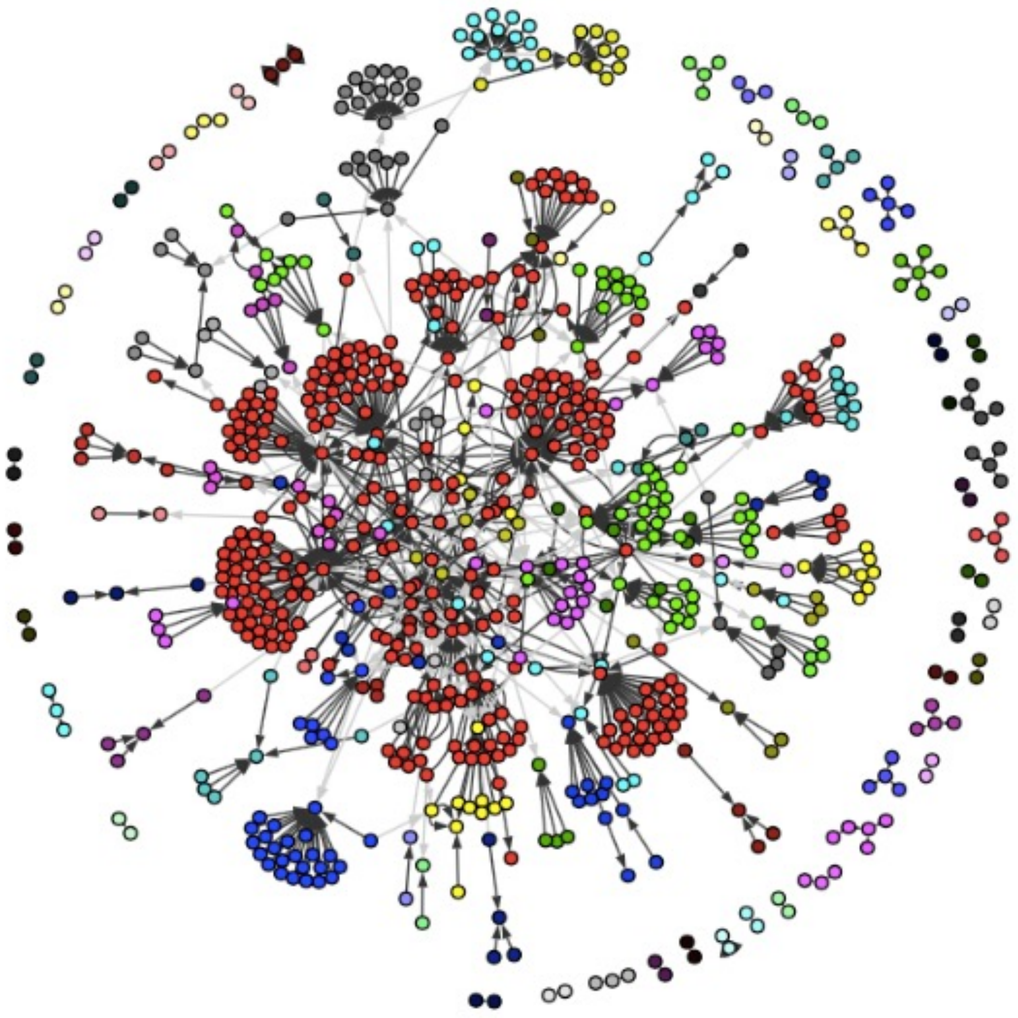
Graph Analytics (1): Link Prediction

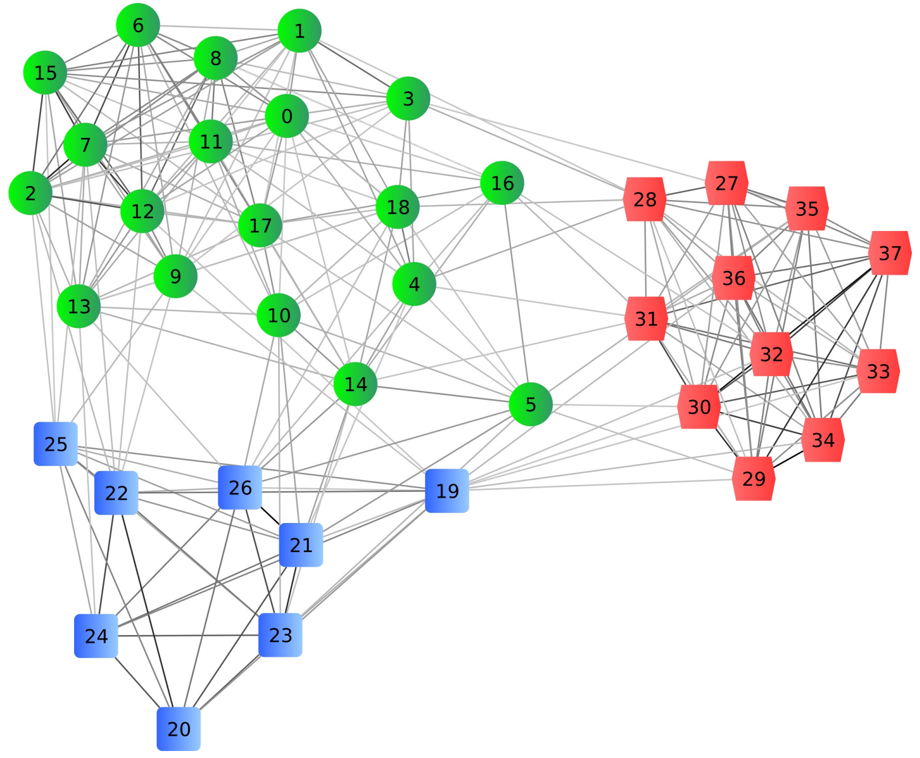


Friend Recommendation: Does Alice Know Bob in Facebook

Item Recommendation: Which Items will The User Like?

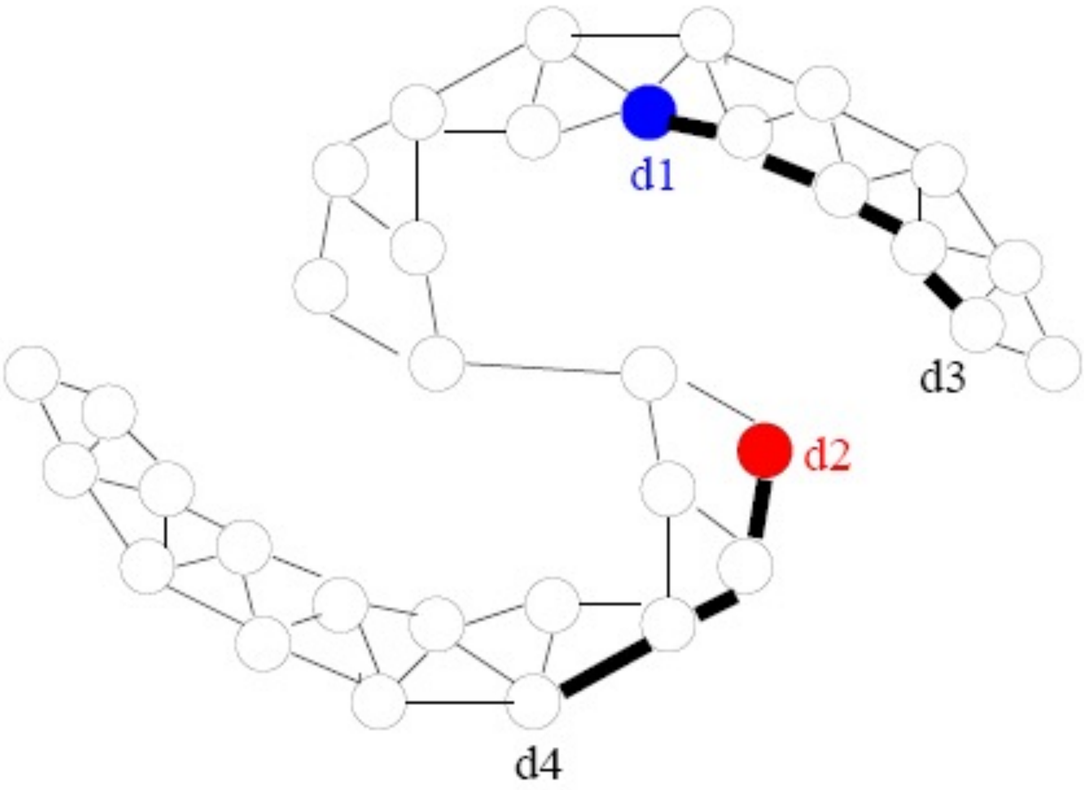
Graph Analytics (2): Community Detection





Typically a Graph Clustering Task

Graph Analytics (3): Node Classification

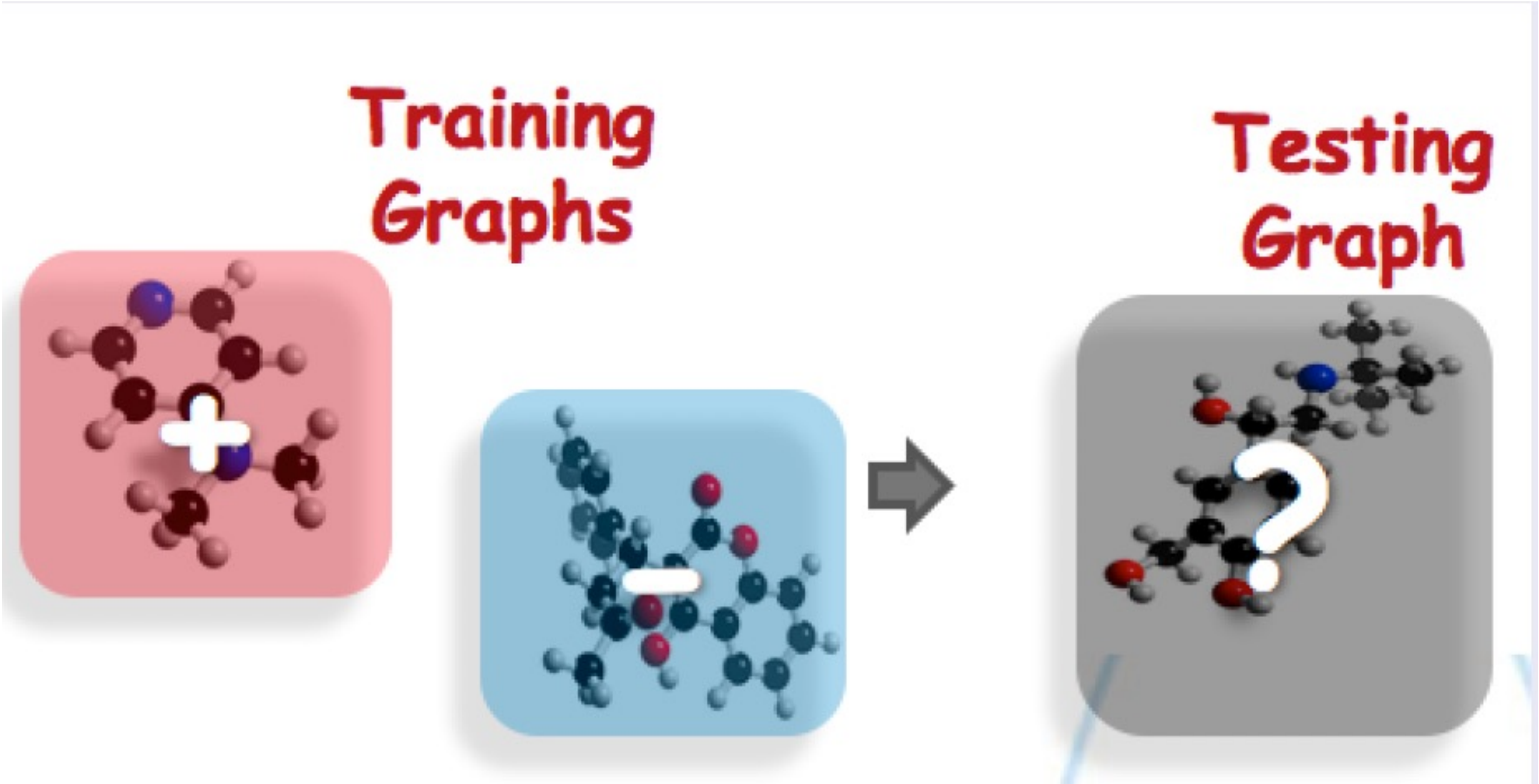


-Graph from Jerry Zhu's Tutorial in ICML 07

- d1 is democratic
- d2 is republican
- What can we say about d3 and d4?

Graph Analytics (4): Graph Classification

- **Example: Drug Activity Prediction in the Biological domain**



It is active to Breast Cancer?

Graph Analytics: Many Others…

- Sampling
- Ranking
- Evolution
- Matching
- Visualization
- Social Influence
- …

Traditional Machine Learning Pipeline

- **Network Feature Extractions**
  - **Nodes: degree/PageRank score**
  - **Edges: # of common neighbors**

- **Feature Vector Construction**
  - Network Feature + Content Feature

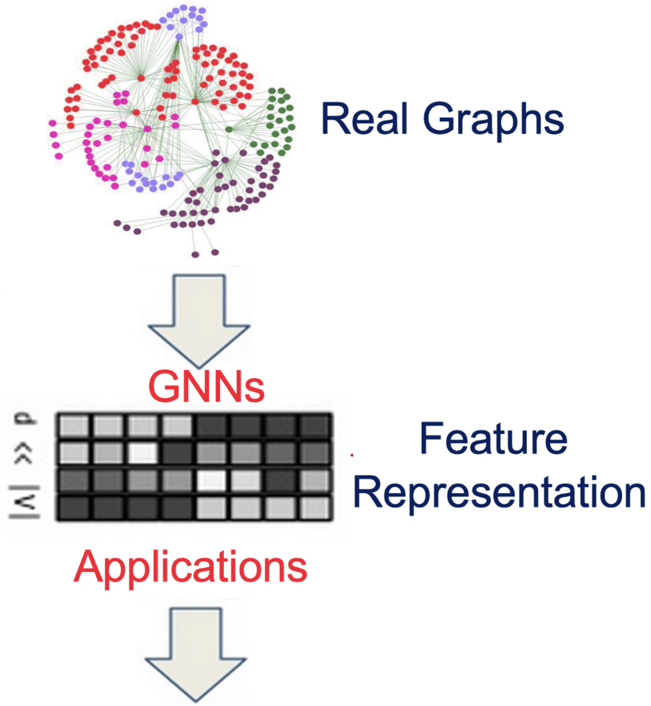
- **Machine Learning Tasks**
  - Classification
  - Clustering
  - Link Prediction

Disadvantages:

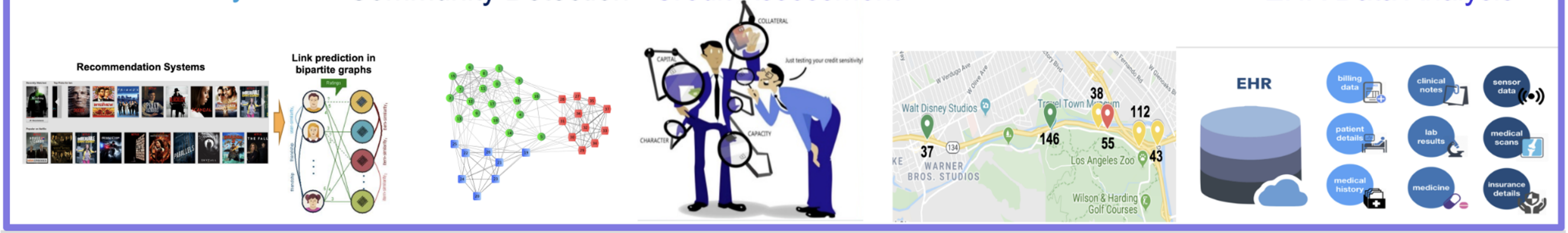
- Ineffective
- Shallow Method
- Multiple Steps

Graph neural networks (GNNs)

- Methods and Applications
  - Frontier of Deep Learning
  - Effective Representation for Graph Data
  - Wide applications



Recommender Systems | Community Detection | Credit Assessment | Traffic Flow Prediction | EHR Data Analysis

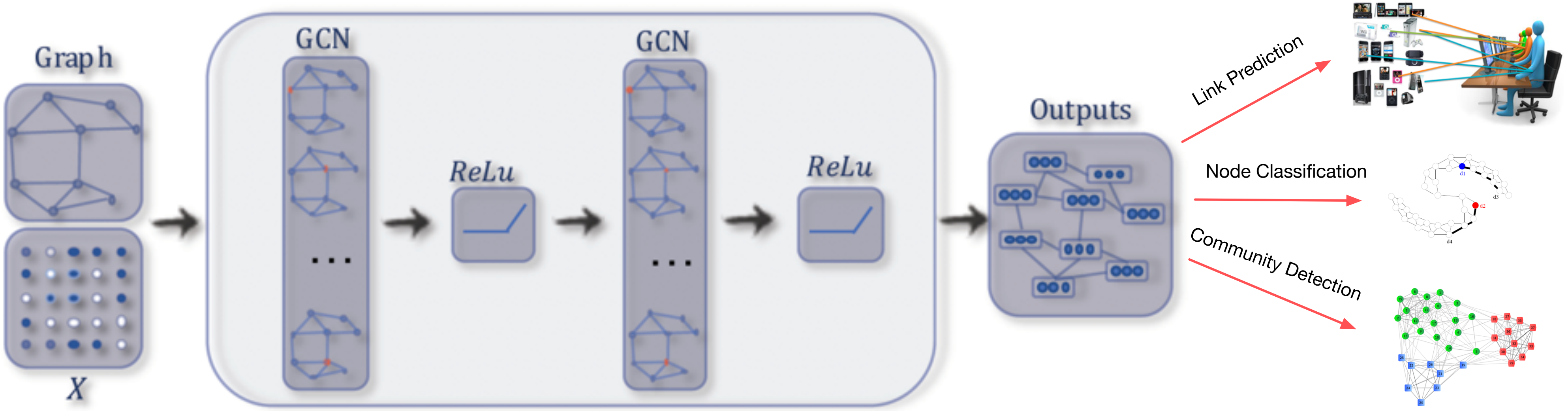


Graph neural networks (GNNs)

A deep encoder which transfer the node in a graph into a latent vector

$$\mathrm{ENC}(v) = \text{multiple layers of non-linear transformations of graph structure}$$

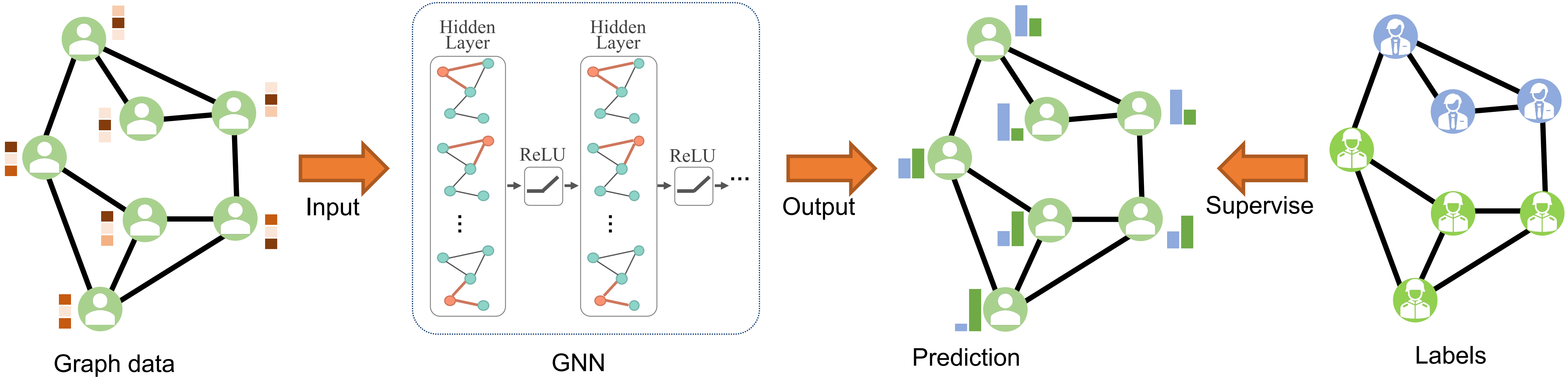
- **Learn Better Representation for Graph Data**



Big Picture of Graph Neural Networks

Motivation of graph self-supervised learning (GSSL)

Recent graph learning focuses on (semi-) supervised learning scenarios…



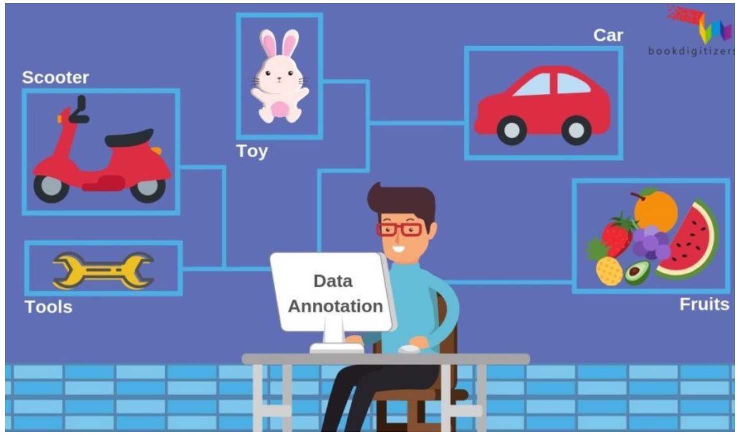
Reliance on **labels**!

Motivation

Recent graph learning focuses on (semi-) supervised learning scenarios…

Reliance on **labels**➔ Problems:

- Problem 1: Expensive cost of data collection and annotation

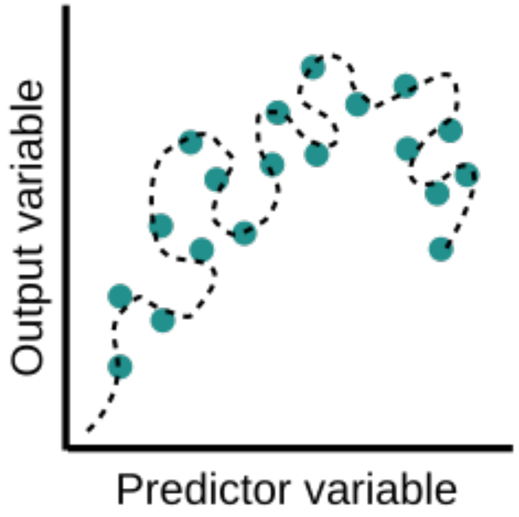


Motivation

Recent graph learning focuses on (semi-) supervised learning scenarios…

Reliance on **labels**➔ Problems:

- Problem 1: Expensive cost of data collection and annotation
- Problem 2: Pool generalization (over-fitting)

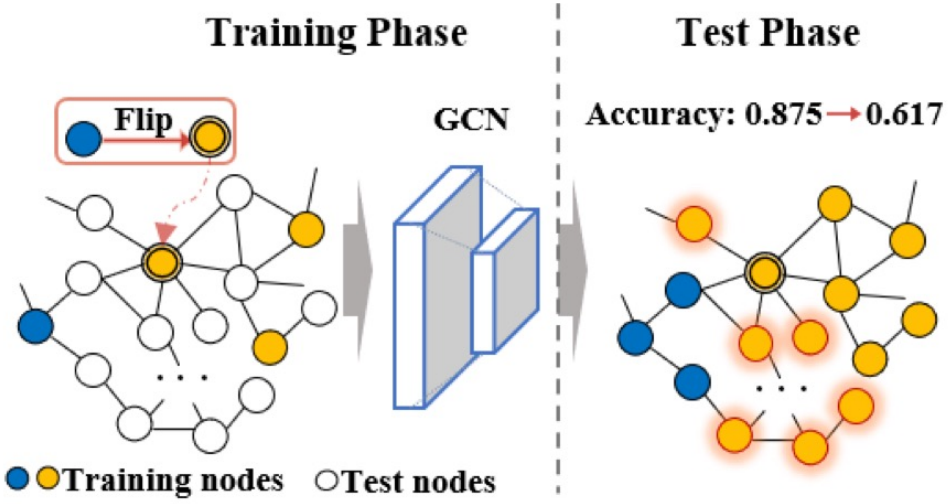


Motivation

Recent graph learning focuses on (semi-) supervised learning scenarios…

Reliance on **labels**➔ Problems:

- Problem 1: Expensive cost of data collection and annotation
- Problem 2: Pool generalization (over-fitting)
- Problem 3: Vulnerable to label-related adversarial attacks



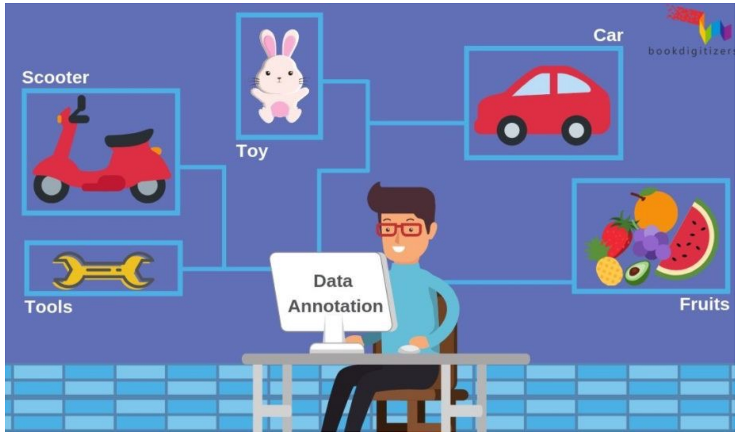
Zhang, M., Hu, L., Shi, C., & Wang, X. (2020). Adversarial Label-Flipping Attack and Defense for Graph Neural Networks. 2020 IEEE International Conference on Data Mining (ICDM), 791-800.

Motivation

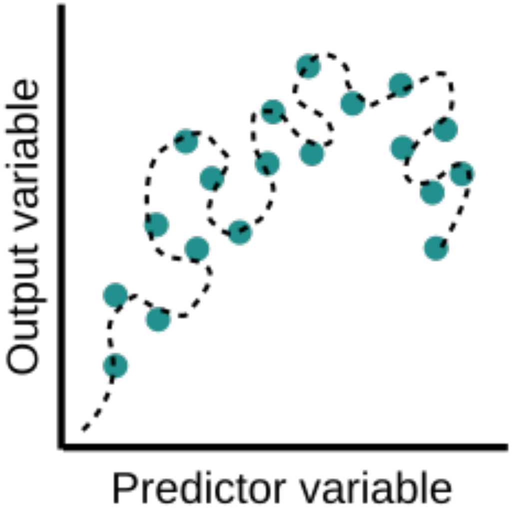
Recent graph learning focuses on (semi-) supervised learning scenarios…

Reliance on **labels**➔ Problems:

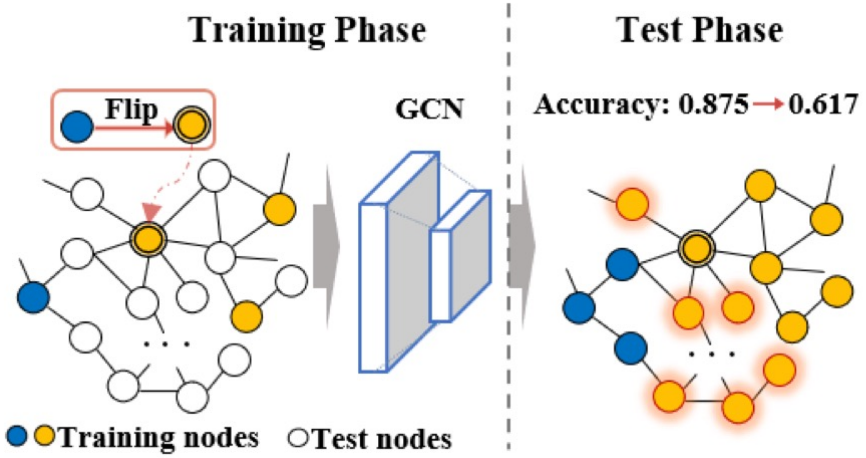
- Expensive cost of data collection and annotation



- Pool generalization



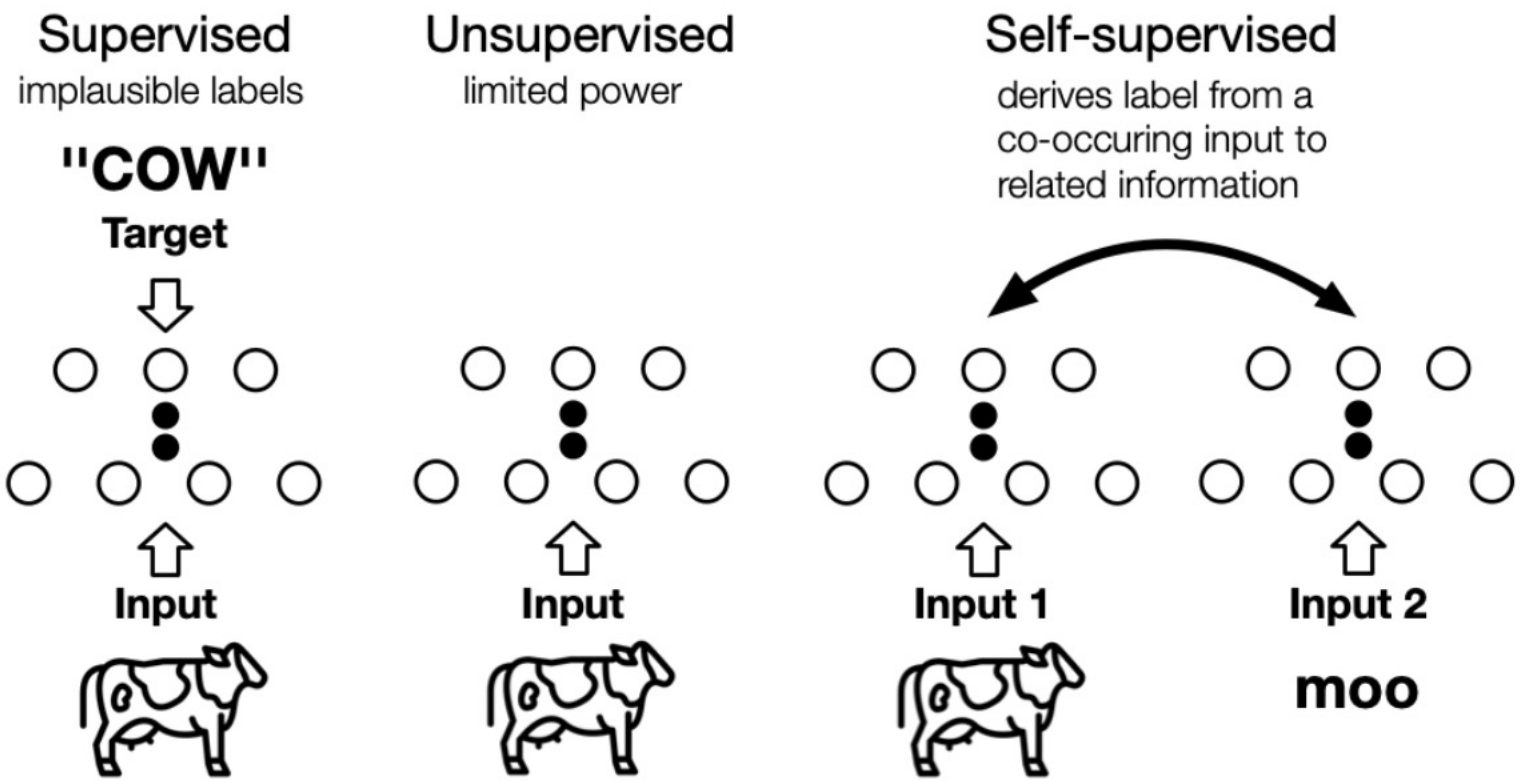
- Vulnerable to label-related adversarial attacks



How to address these problems?

Zhang, M., Hu, L., Shi, C., & Wang, X. (2020). Adversarial Label-Flipping Attack and Defense for Graph Neural Networks. 2020 IEEE International Conference on Data Mining (ICDM), 791-800.

Self-supervised Learning (SSL)

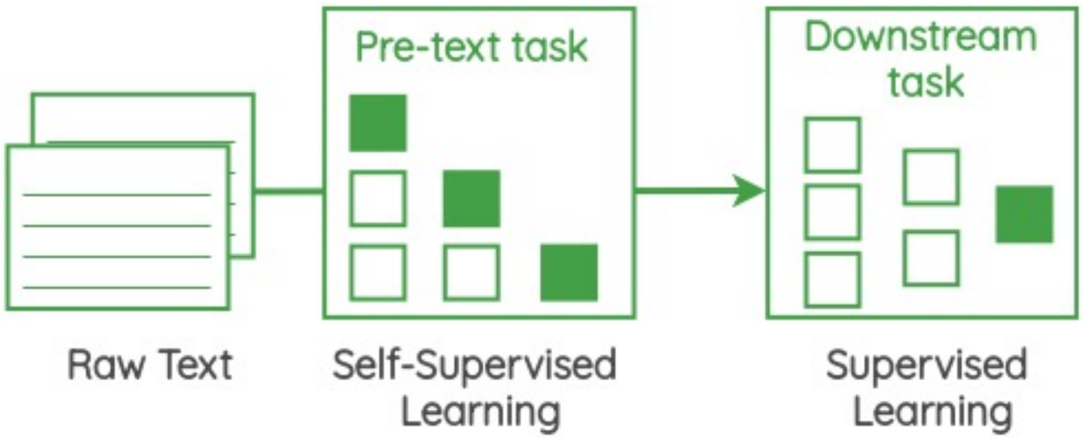


Instead of relying on human-annotated labels, self-supervised learning acquires “labels” from data itself by using an “automatic” process.

Reduces the dependence on manual labels!

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering.

Self-supervised Learning (SSL)



"**pretext task**": use the data itself to generate labels and use supervised methods to solve unsupervised problems.

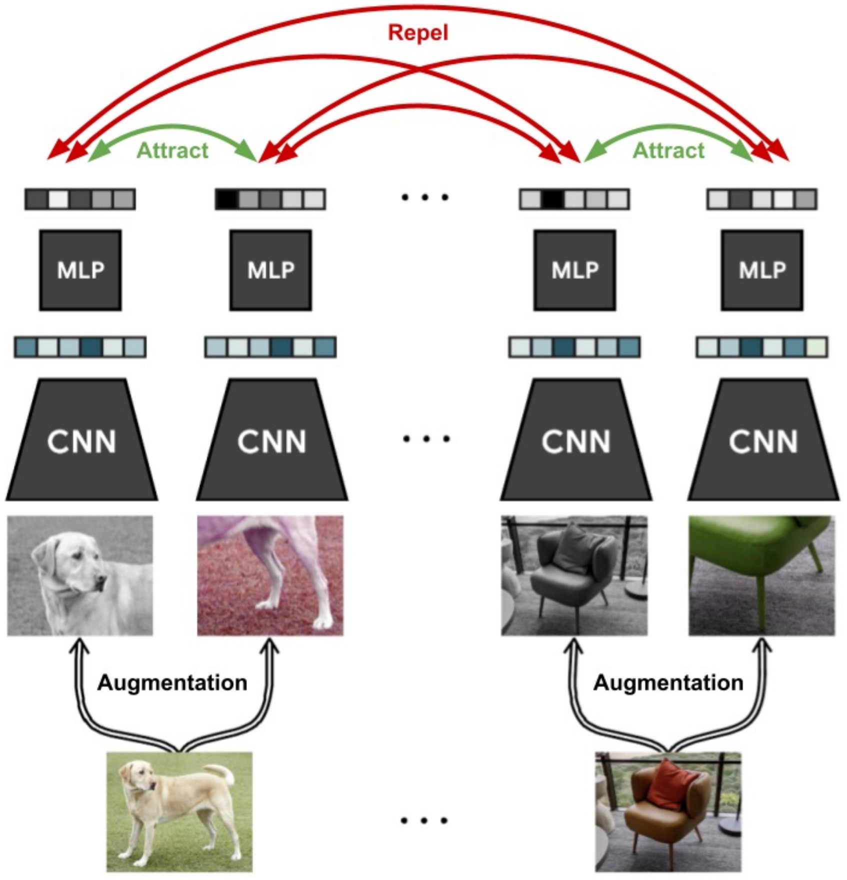
The representations learned by performing this task can be used as a starting point for our **downstream supervised tasks**.

Critical problem: how to design the pretext task?

https://amitness.com/2020/05/self-supervised-learning-nlp/

Self-supervised Learning: Computer Vision

Contrastive learning:



SimCLR

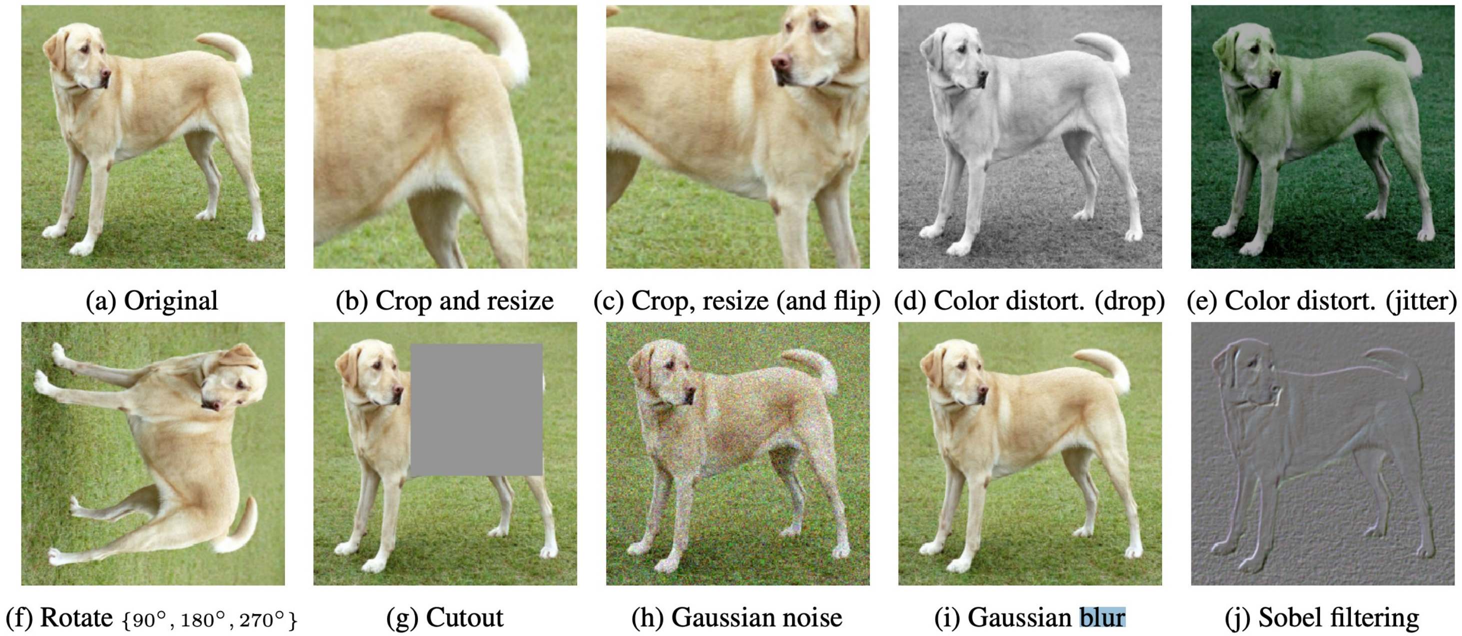


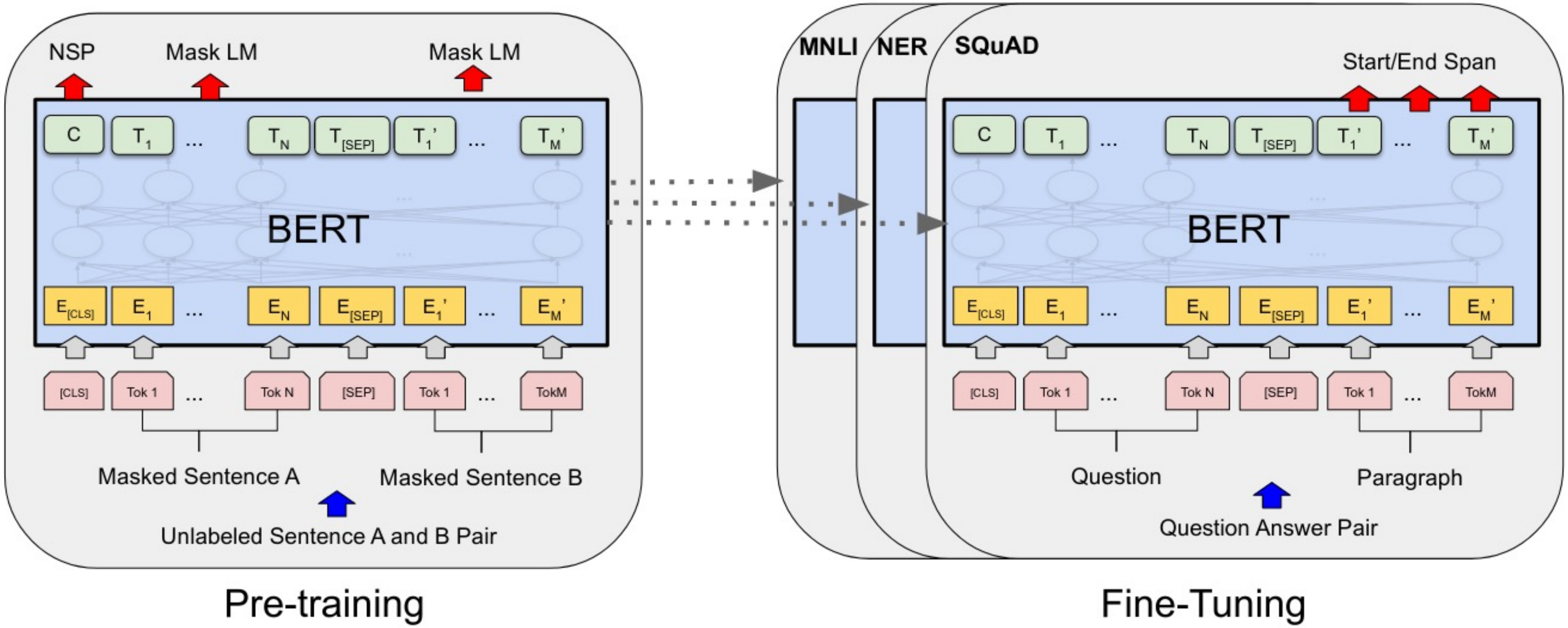
Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

https://simclr.github.io

https://www.casualganpapers.com/self-supervised-contrastive-representation-learning/BYOL-explained.html

Self-supervised Learning: NLP

Large-scale pre-trained language model: BERT



2 self-supervised pre-training schemes of BERT:

- Masked Language Modeling (MLM)
- Next Sentence Prediction (NSP)

https://medium.com/swlh/bert-pre-training-of-transformers-for-language-understanding-5214fba4a9af

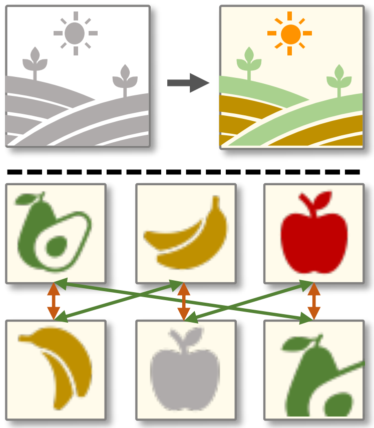
Self-supervised Learning on graphs

How to **design pretext tasks** in graph domain?

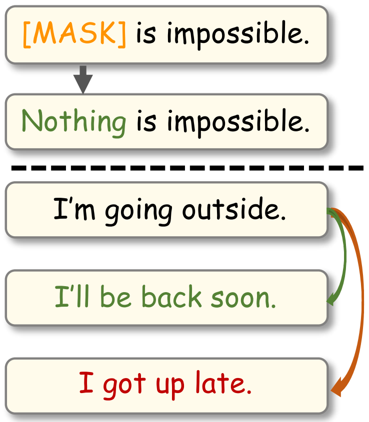
Can we transfer the pretext tasks designed for CV/NLP to graph domain?
- Not trivial!

Data space
- CV/NLP: 2D/1D regular-grid Euclidean space
- Graph: Non-Euclidean space

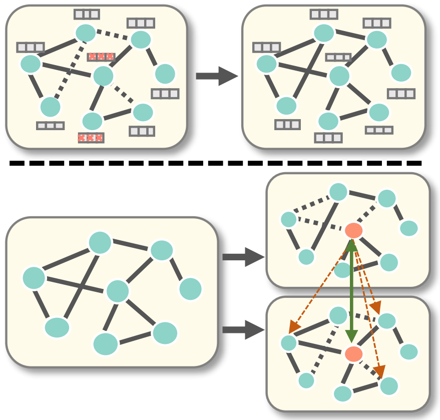
Reliance between samples
- CV/NLP: Independent samples (image/text)
- Graph: data examples (nodes) in graph data are correlated by the topological structure



SSL on CV



SSL on NLP



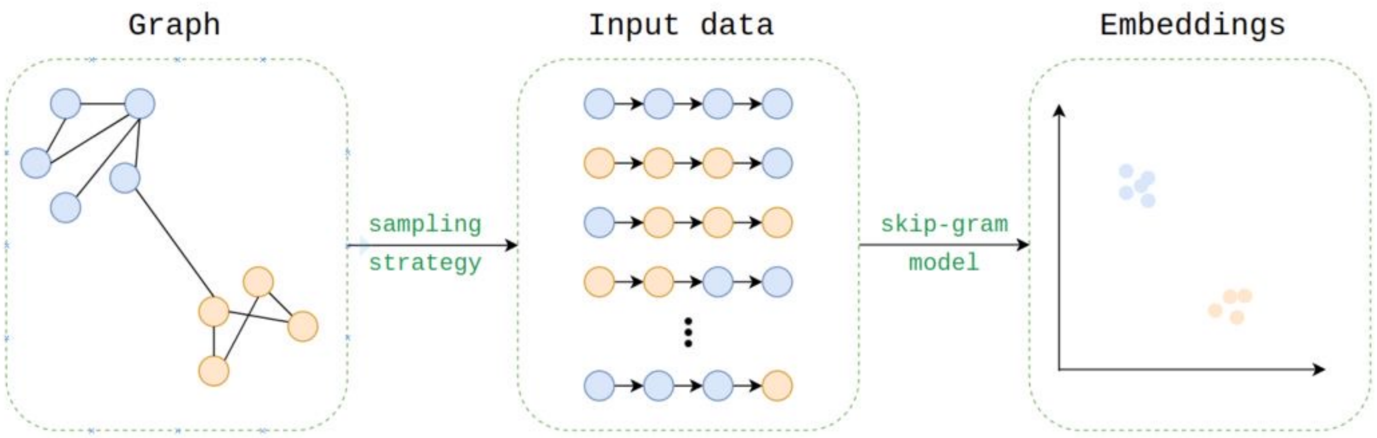
SSL on graph

Cannot easily transfer!
Need: exclusive definitions and taxonomies

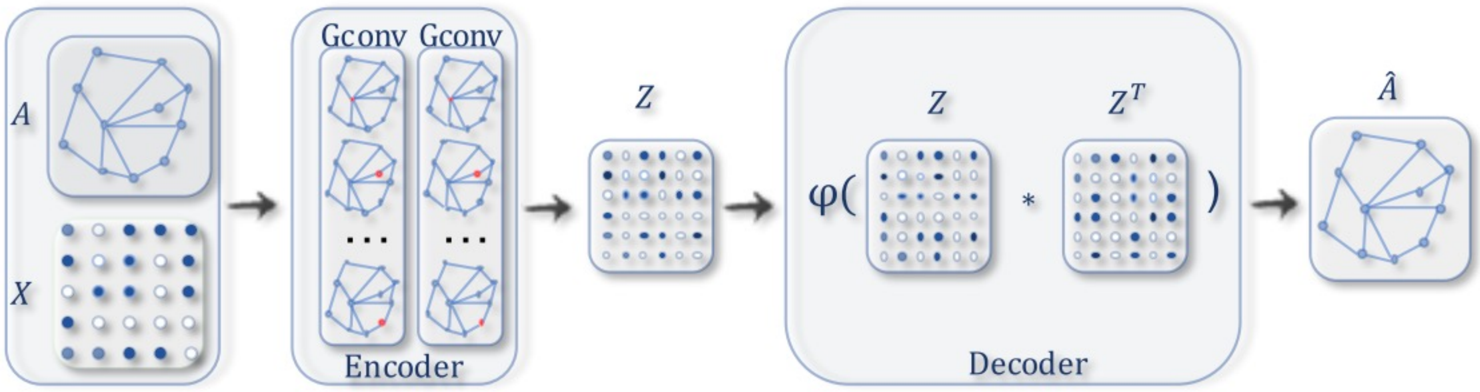
Self-supervised Learning on graphs

Early studies:

- Node2vec:



- Graph autoencoder (GAE)



https://maelfabien.github.io/machinelearning/graph_5/

Self-supervised Learning on graphs

A pioneer work of graph SSL:

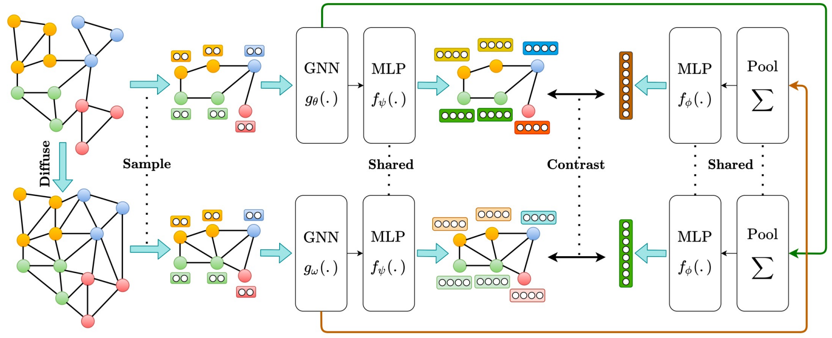
Deep Graph Infomax

Authors	Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, R Devon Hjelm
Publication date	2019/5
Journal	7th International Conference on Learning Representations (ICLR 2019)
Total citations	Cited by 1535

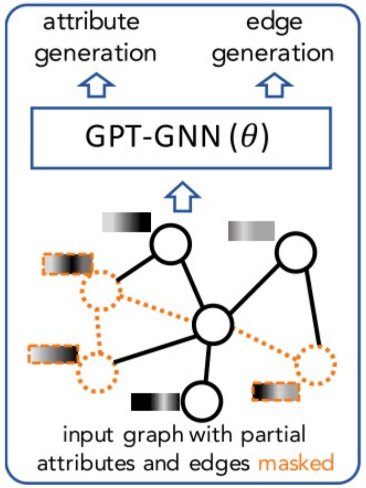
Growing trend!

Self-supervised Learning on graphs

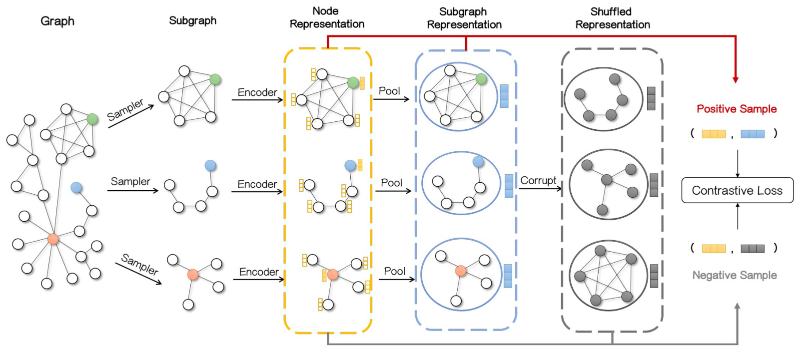
After DGI…



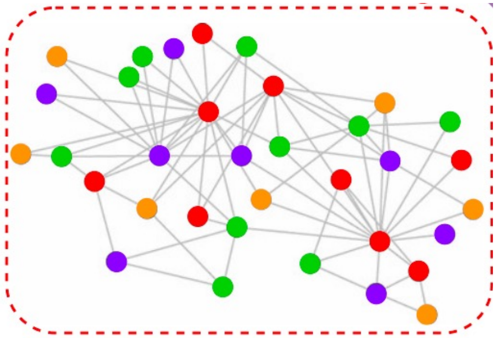
Multi-view contrastive learning[1]



Graph generation [2]



Subgraph contrastive learning [3]



Clustering prediction[4]

Following questions:

- Which are the representative works?
- How to categorize them?
- How to formulate them with a unified framework?
- What is the research frontiers?
- Where can GSSL be applied?
- What are the potential future directions?

[1] Hassani, K., & Khasahmadi, A. H. (2020, November). Contrastive multi-view representation learning on graphs. In International Conference on Machine Learning (pp. 4116-4126). PMLR.
[2] Hu, Z., Dong, Y., Wang, K., Chang, K. W., & Sun, Y. (2020, August). Gpt-gnn: Generative pre-training of graph neural networks. In Proceedings of the 26th ACM SIGKDD (pp. 1857-1867).
[3] Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., & Zhu, Y. (2020, November). Sub-graph contrast for scalable self-supervised graph representation learning. In 2020 IEEE ICDM (pp. 222-231). IEEE.
[4] You, Y., Chen, T., Wang, Z., & Shen, Y. (2020, November). When does self-supervision help graph convolutional networks?. In International Conference on Machine Learning (pp. 10871-10880). PMLR.

Part 2:Taxonomy of graph self-supervised learning

- Uniform framework
- Categories of GSSL
- Representative methods

Graph Self-Supervised Learning: A Survey

IEEE TKDE-2022

Graph Self-Supervised Learning: A Survey

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, Philip S. Yu, *Life Fellow, IEEE*

Abstract—Deep learning on graphs has attracted significant interests recently. However, most of the works have focused on (semi-) supervised learning, resulting in shortcomings including heavy label reliance, poor generalization, and weak robustness. To address these issues, self-supervised learning (SSL), which extracts informative knowledge through well-designed pretext tasks without relying on manual labels, has become a promising and trending learning paradigm for graph data. Different from SSL on other domains like computer vision and natural language processing, SSL on graphs has an exclusive background, design ideas, and taxonomies. Under the umbrella of *graph self-supervised learning*, we present a timely and comprehensive review of the existing approaches which employ SSL techniques for graph data. We construct a unified framework that mathematically formalizes the paradigm of graph SSL. According to the objectives of pretext tasks, we divide these approaches into four categories: generation-based, auxiliary property-based, contrast-based, and hybrid approaches. We further describe the applications of graph SSL across various research fields and summarize the commonly used datasets, evaluation benchmark, performance comparison and open-source codes of graph SSL. Finally, we discuss the remaining challenges and potential future directions in this research field.

Index Terms—Self-supervised learning, graph analytics, deep learning, graph representation learning, graph neural networks.

Graph self-supervised learning: A survey

Y Liu, M Jin, S Pan, C Zhou, Y Zheng… - … on Knowledge and …, 2022 - ieeexplore.ieee.org

Deep learning on graphs has attracted significant interests recently. However, most of the works have focused on (semi-) supervised learning, resulting in shortcomings including heavy label reliance, poor generalization, and weak robustness. To address these issues, self-supervised learning (SSL), which extracts informative knowledge through well-designed pretext tasks without relying on manual labels, has become a promising and trending learning paradigm for graph data. Different from SSL on other domains like computer vision …

https://arxiv.org/abs/2103.00111

Overview

- **Unified framework and systematic taxonomy**

We propose a unified framework that mathematically formalizes graph SSL approaches. Based on our framework, we systematically categorize the existing works into four categories.

- **Comprehensive and up-to-date review**

We conduct a comprehensive and timely review for classical and latest graph SSL approaches.

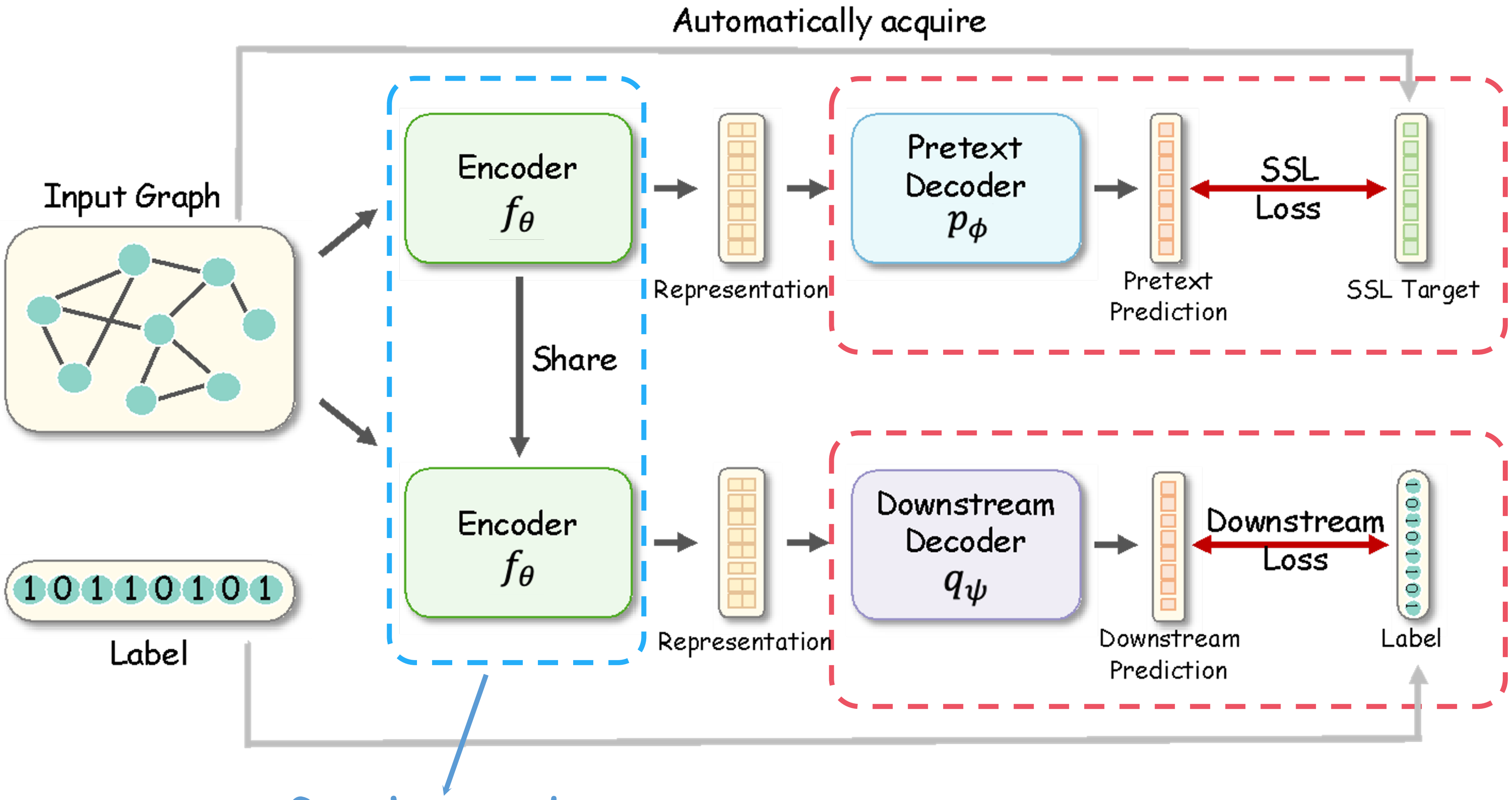
- **Abundant resources and applications.**

We collect abundant resources on graph SSL, including datasets, evaluation benchmark, performance comparison, and open-source codes. We also summarize the practical applications of graph SSL in various research fields.

- **Outlook on future directions**

We point out the technical limitations of current research. We further suggest six promising directions for future works from different perspectives.

Encoder-Decoder Framework

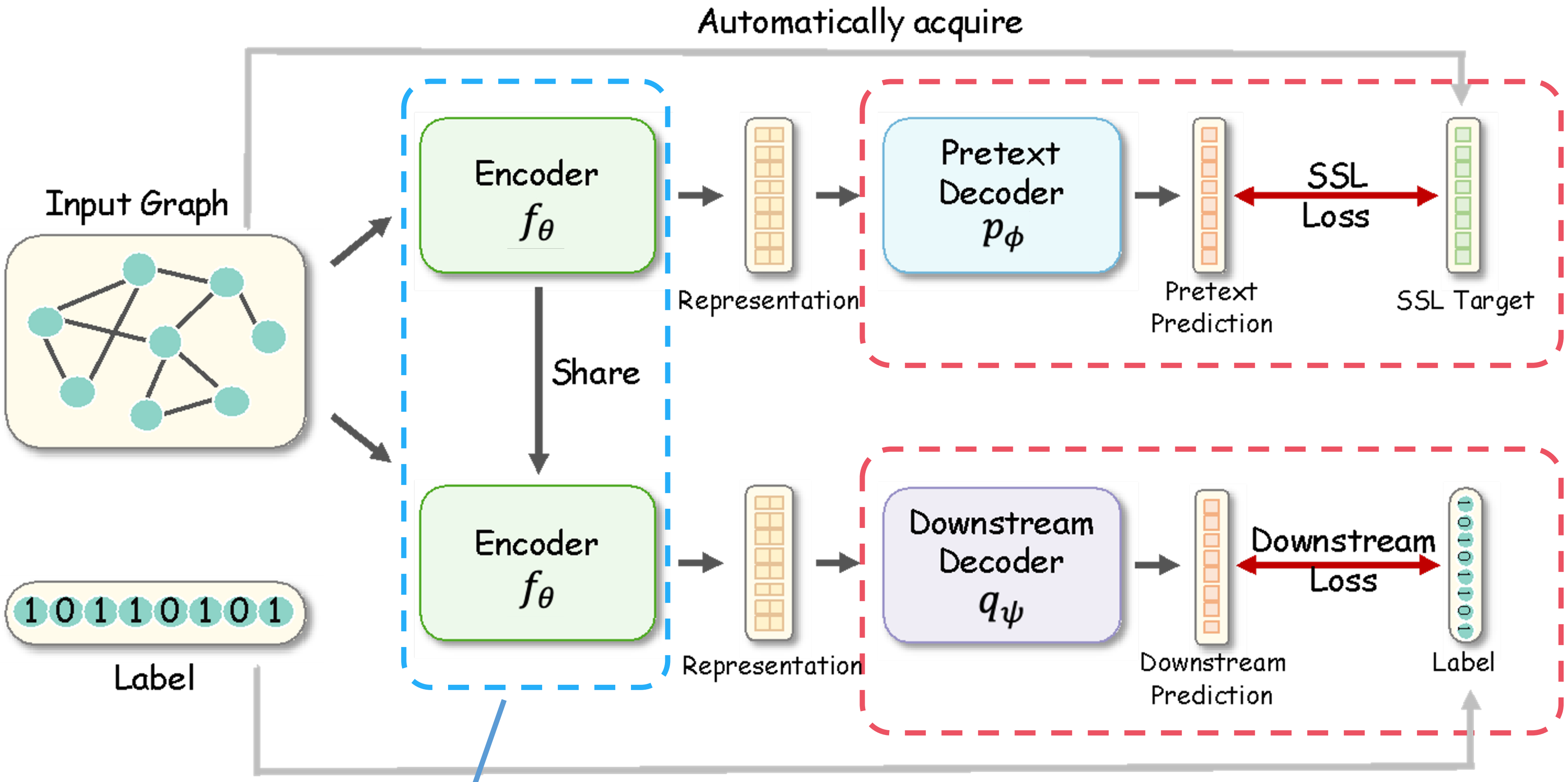


Pretext task:
For encoder training

Downstream task:
To solve real-world problems

Graph encoder:
GNNs, Transformers, DNN…

Encoder-Decoder Framework



Pretext task:
For encoder training

Q2: Which types of pretext tasks do we have?

Downstream task:
To solve real-world problems

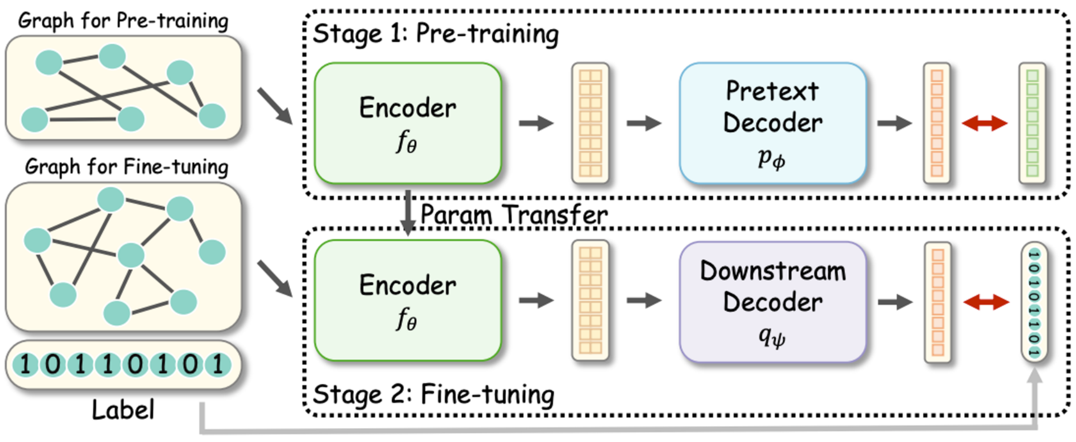
Q3: What kind of downstream tasks can be solved?

Graph encoder:
GNNs, Transformers, DNN…

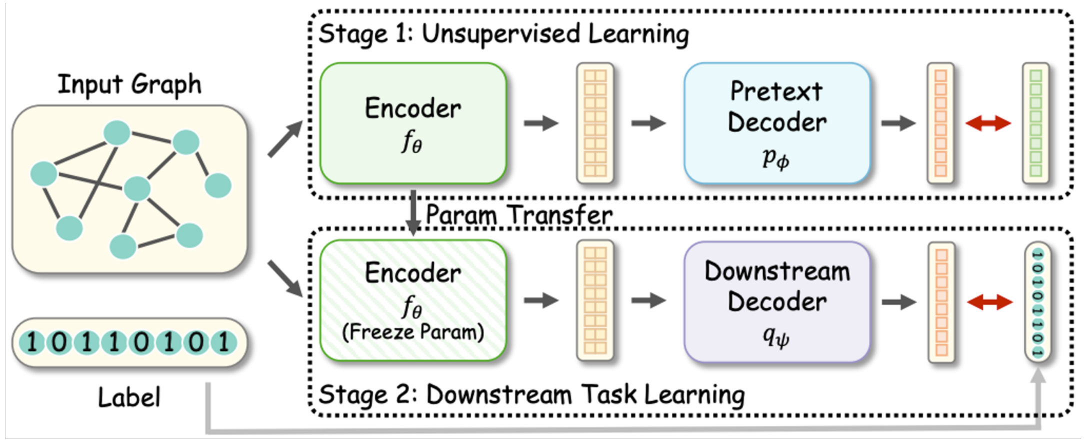
Q1: How to share the encoder between two tasks?

3 SSL schemes

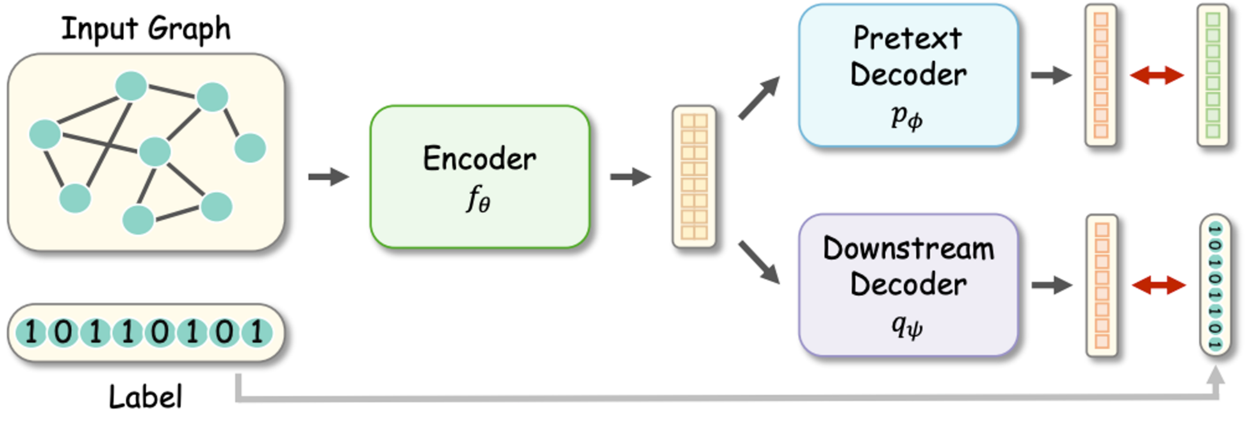
Q1: How to share the encoder between two tasks?



(i) Pre-training and Fine-tuning (PF)



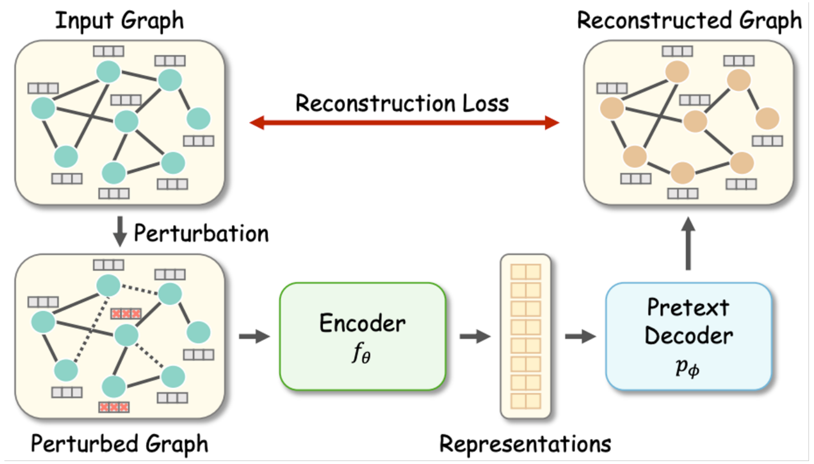
(iii) Unsupervised Representation Learning (URL)



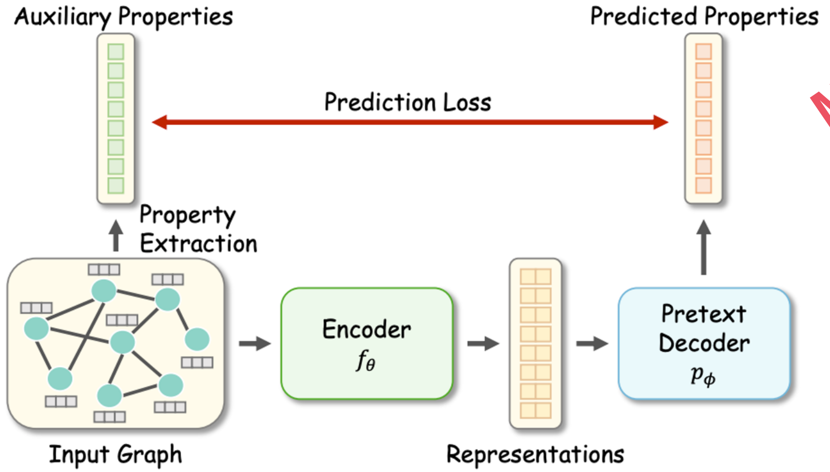
(ii) Joint Learning (JL)

4 Categories of Graph SSL

Q2: Which types of pretext tasks do we have?

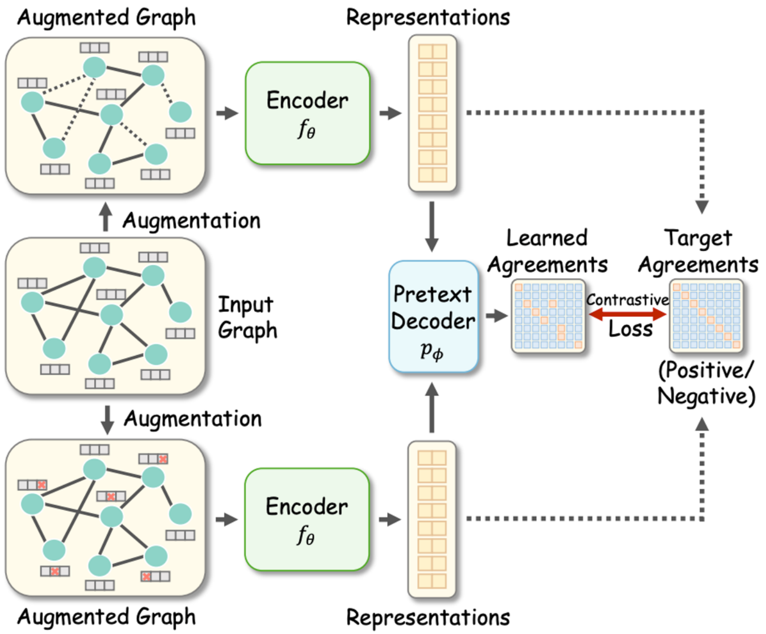


(i) Generation-based

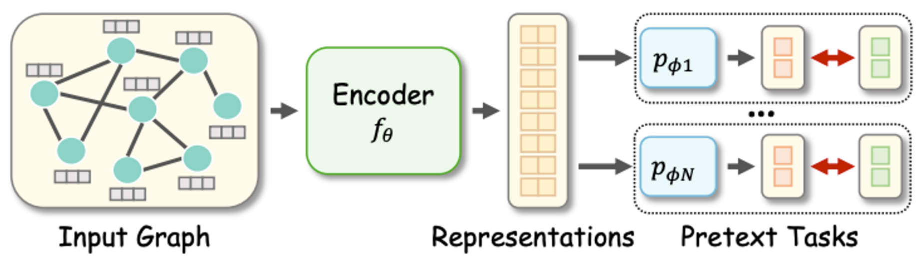


(ii) Auxiliary Property-based

Main Taxonomy!



(iii) Contrast-based



(iv) Hybrid

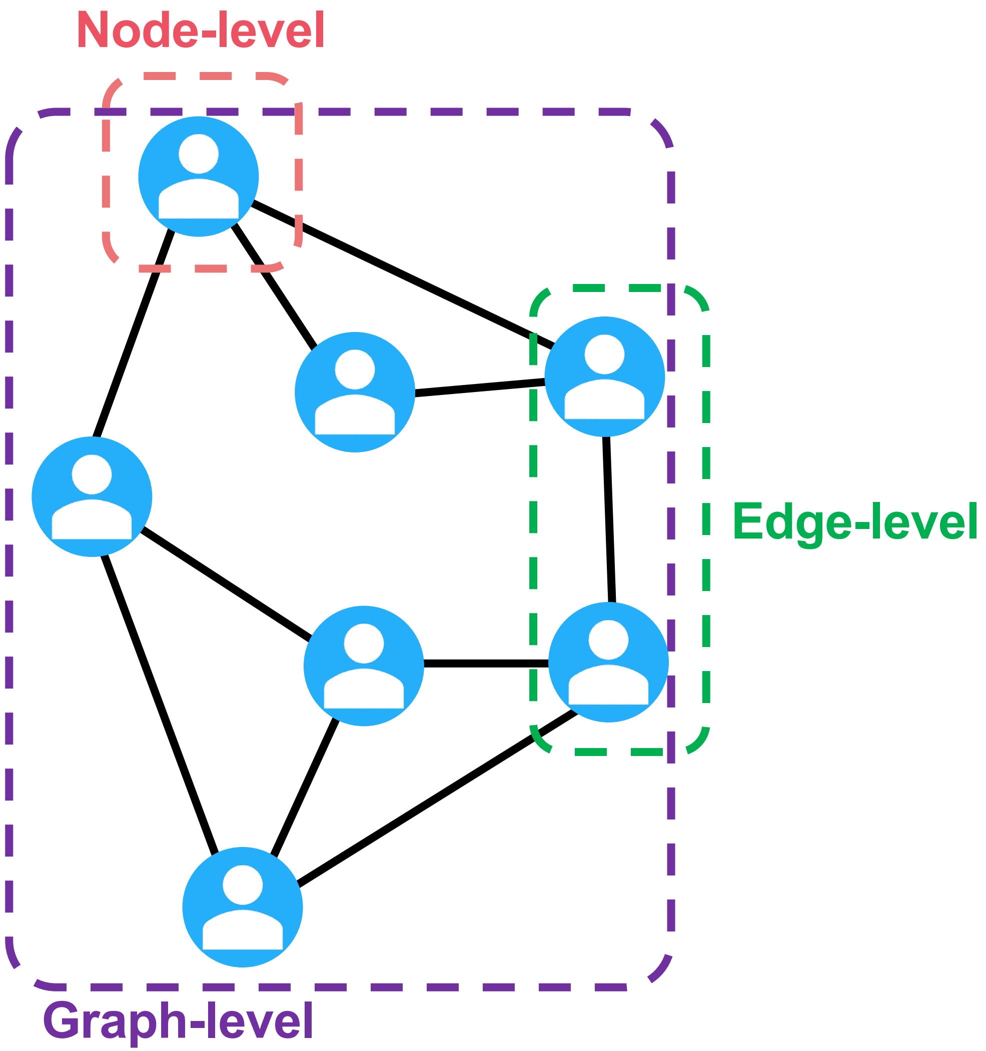
3 Types of Downstream Tasks

Q3: What kind of downstream tasks can be solved?

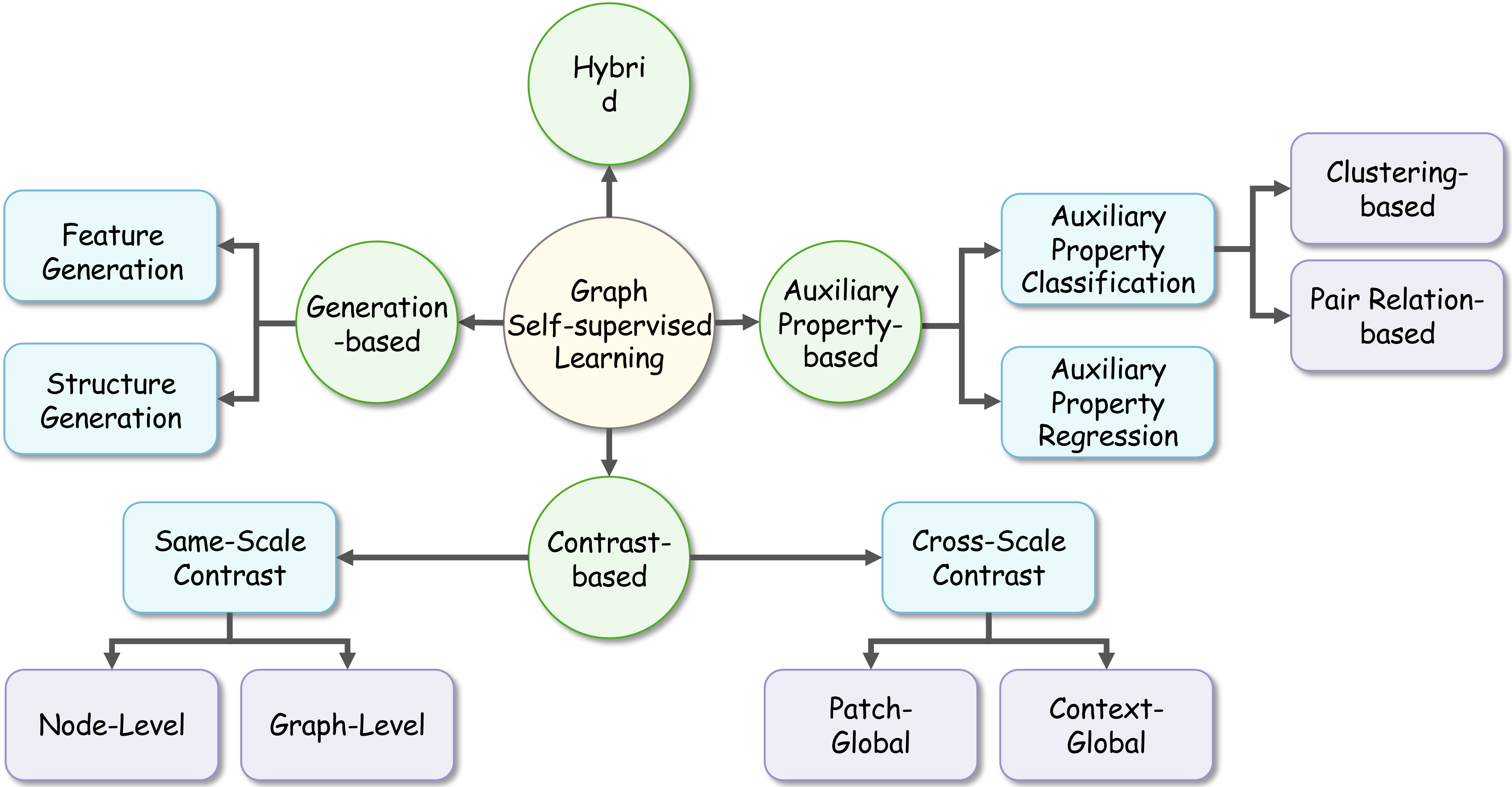
(i) Node-level tasks:
Node classification, node regression…

(ii) Edge-level tasks:
Link prediction, edge classification…

(iii) Graph-level tasks:
graph classification, graph regression, …



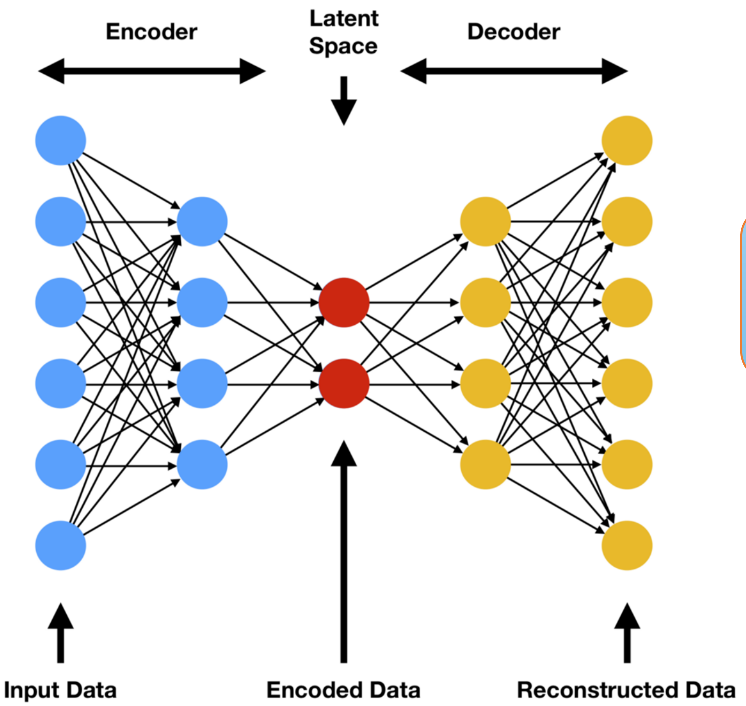
Outline of Graph SSL



Generation-based Methods: Origin

Generation-based methods aim to reconstruct the input data and use the input data as the supervision signals.

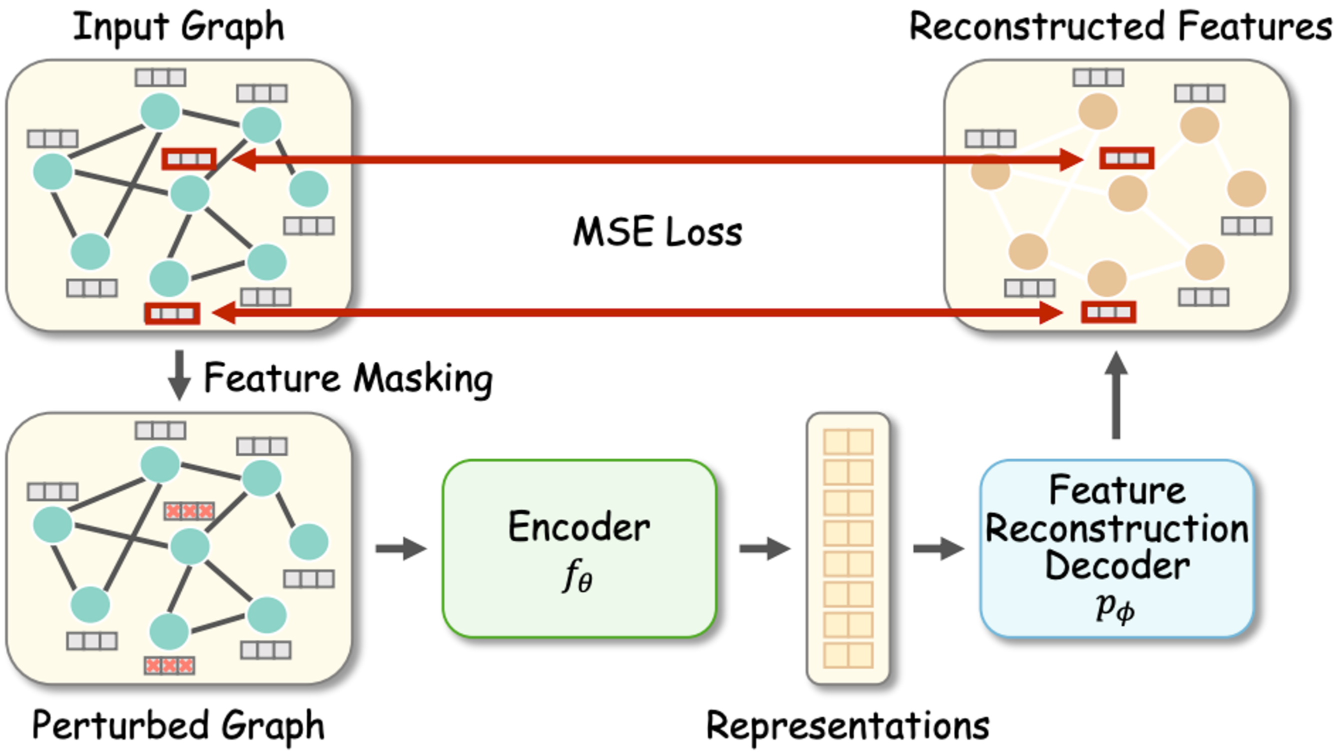
Origin: Autoencoder



Generation-based Methods

- **Feature Generation**: reconstruct the feature information
- **Structure Generation**: reconstruct the topological structure information

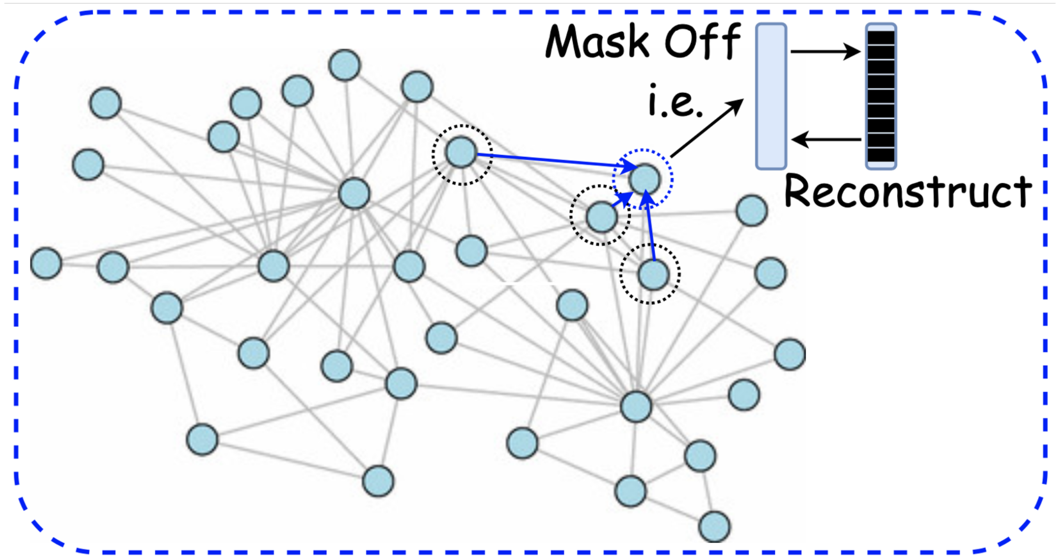
Feature Generation

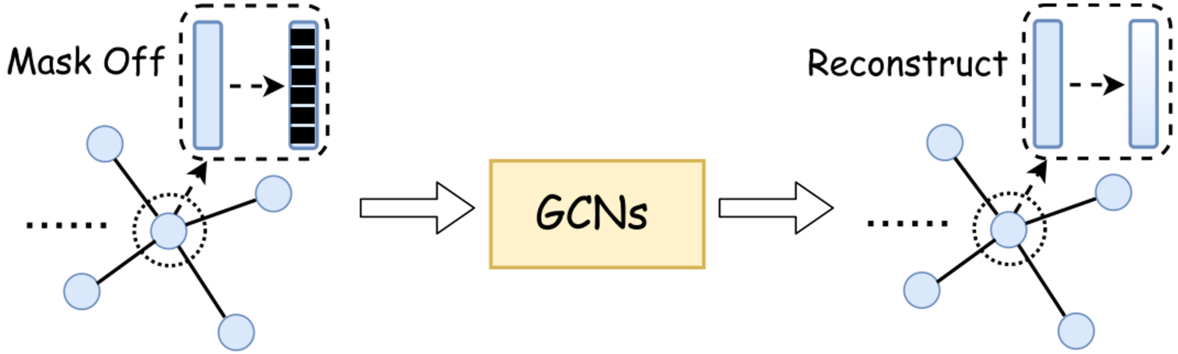


- **Pretext Decoder:** Fully connected layers that regresses the features
- **SSL Loss:** Regression loss (MSE)

Feature Generation: Representative Method

- **Graph completion**



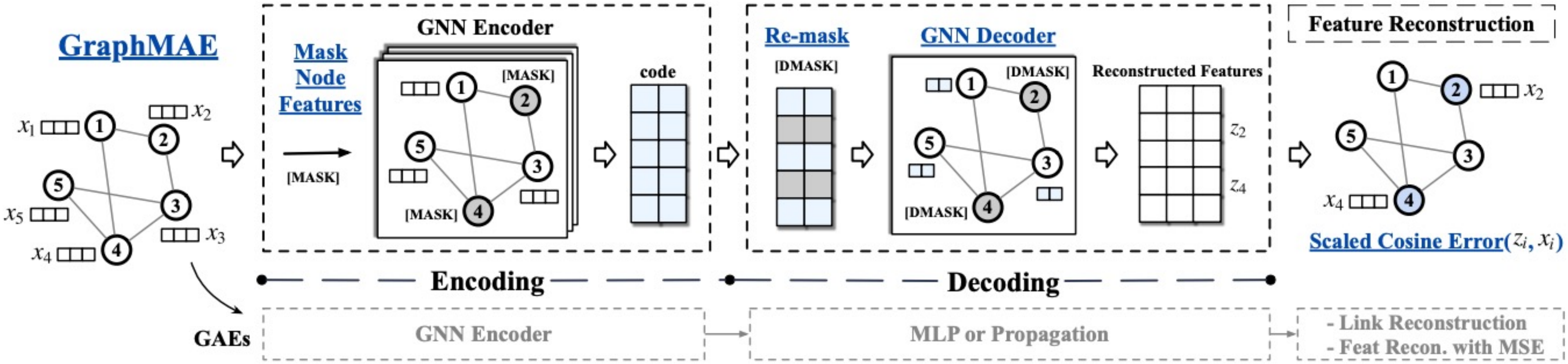


Intuition: Use the neighboring information to reconstruct the masked features (similar to MLM in BERT)

You, Y., Chen, T., Wang, Z., & Shen, Y. (2020, November). When does self-supervision help graph convolutional networks?. In International Conference on Machine Learning (pp. 10871-10880). PMLR.

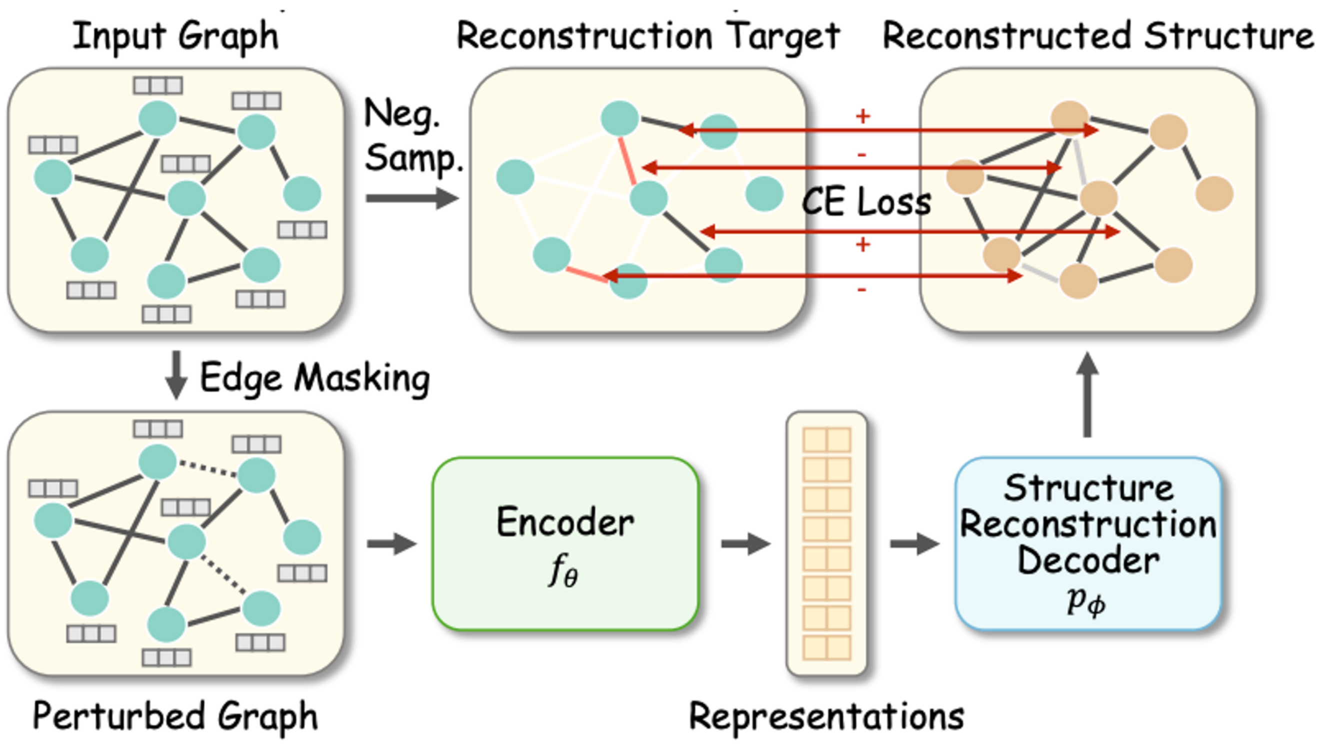
Feature Generation: Representative Method

- **Self-Supervised Masked Graph Autoencoder (GraphMAE)**



Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., & Tang, J. (2022, August). Graphmae: Self-supervised masked graph autoencoders. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 594-604).

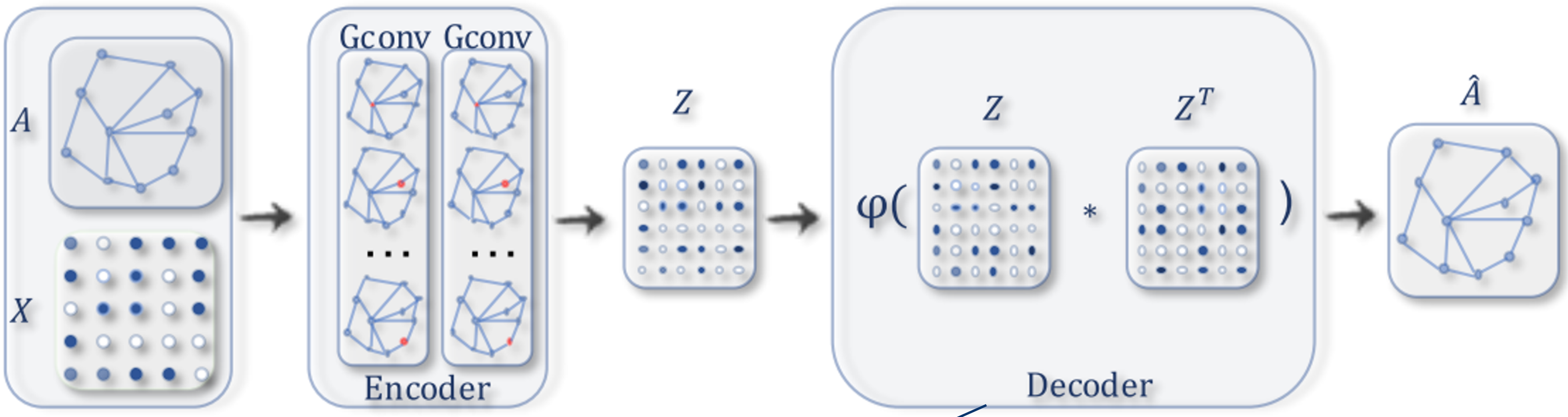
Structure Generation



- **Pretext Decoder:** Adjacency matrix reconstruction network
- **SSL Loss:** Binary cross-entropy

Structure Generation: Representative Method

- **Graph Autoencoder (GAE)**

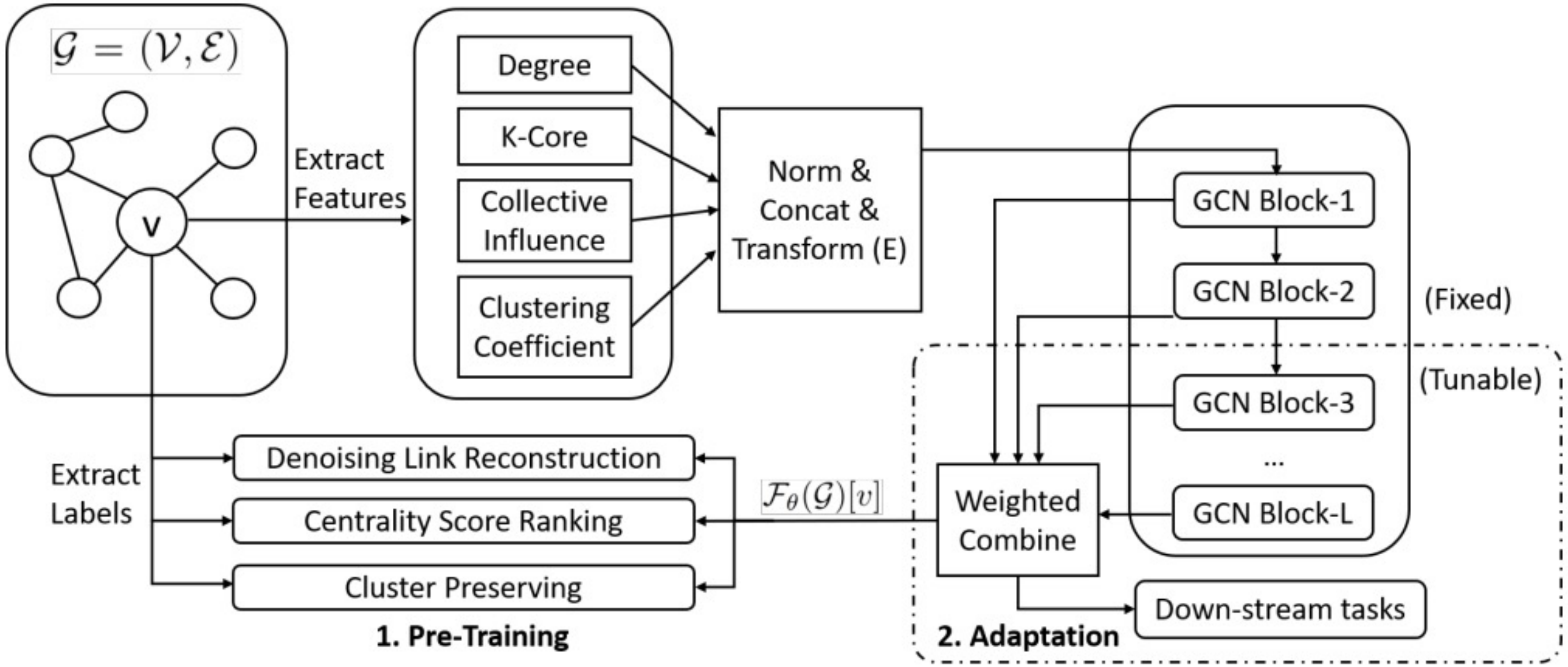


Dot production decoder

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. NeurIPS Workshop.

Structure Generation: Representative Method

- **Pre-Training GNNs for Generic Structural Feature Extraction**

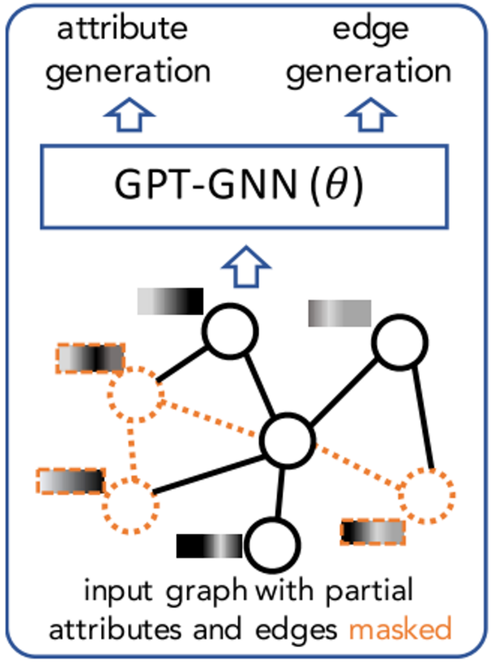


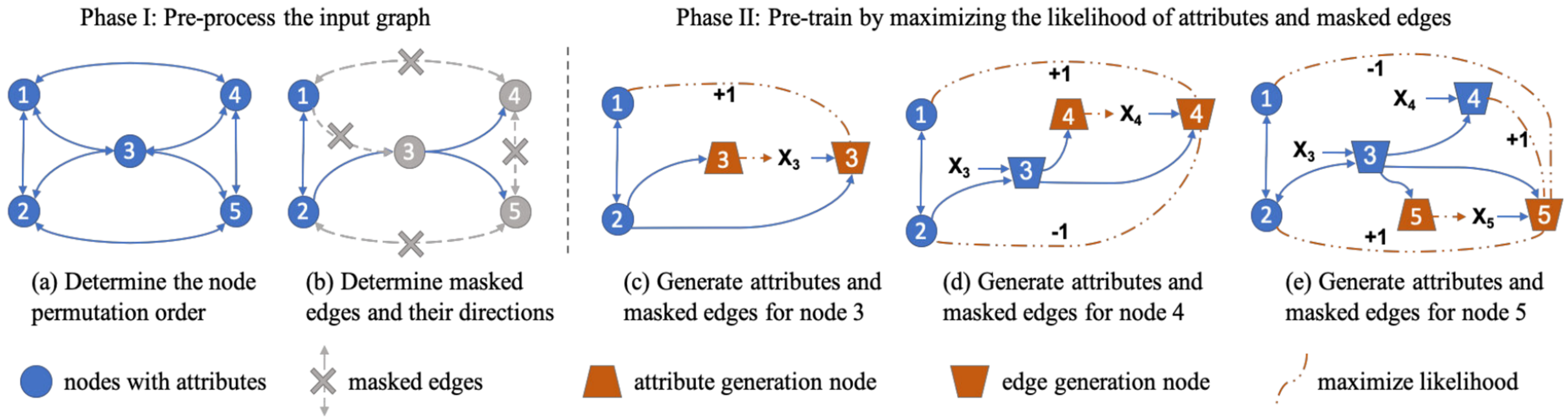
- A multi-layer GNN is pre-trained on three structure-guided tasks
- Part of GNN layers are fine-tuned on the given downstream tasks

Z. Hu, C. Fan, T. Chen, K.-W. Chang, and Y. Sun, “Pre-training graph neural networks for generic structural feature extraction,” arXiv:1905.13728, 2019.

Other Representative Generation-Based Methods

- GPT-GNN





Feature generation + Structure generation

Hu, Z., Dong, Y., Wang, K., Chang, K. W., & Sun, Y. (2020, August). Gpt-gnn: Generative pre-training of graph neural networks. In Proceedings of the 26th ACM SIGKDD (pp. 1857-1867).

Generation-based Methods: Summary

Approach	Pretext Task Category	Downstream Task Level	Training Scheme	Data Type of Graph	Input Data Perturbation	Generation Target
Graph Completion [17]	FG	Node	PF/JL	Attributed	Feature Masking	Node Feature
AttributeMask [40]	FG	Node	PF/JL	Attributed	Feature Masking	PCA Node Feature
AttrMasking [16]	FG	Node	PF	Attributed	Feature Masking	Node/Edge Feature
MGAE [41]	FG	Node	JL	Attributed	Feature Noising	Node Feature
Corrupted Features Reconstruction [42]	FG	Node	JL	Attributed	Feature Noising	Node Feature
Corrupted Embeddings Reconstruction [42]	FG	Node	JL	Attributed	Embedding Noising	Node Embedding
GALA [43]	FG	Node/Link	JL	Attributed	-	Node Feature
Autoencoding [42]	FG	Node	JL	Attributed	-	Node Feature
GAE/VGAE [32]	SG	Link	URL	Attributed	-	Adjacency Matrix
SIG-VAE [44]	SG	Node/Link	URL	Plain/Attributed	-	Adjacency Matrix
ARGA/ARVGA [45]	SG	Node/Link	URL	Attributed	-	Adjacency Matrix
SuperGAT [46]	SG	Node	JL	Attributed	-	Partial Edge
Denoising Link Reconstruction [47]	SG	Node/Link/Graph	PF	Attributed	Edge Masking	Masked Edge
EdgeMask [40]	SG	Node	PF/JL	Attributed	Edge Masking	Masked Edge
Zhu et al. [48]	SG	Node	PF	Attributed	Feature Masking/Edge Masking	Partial Edge

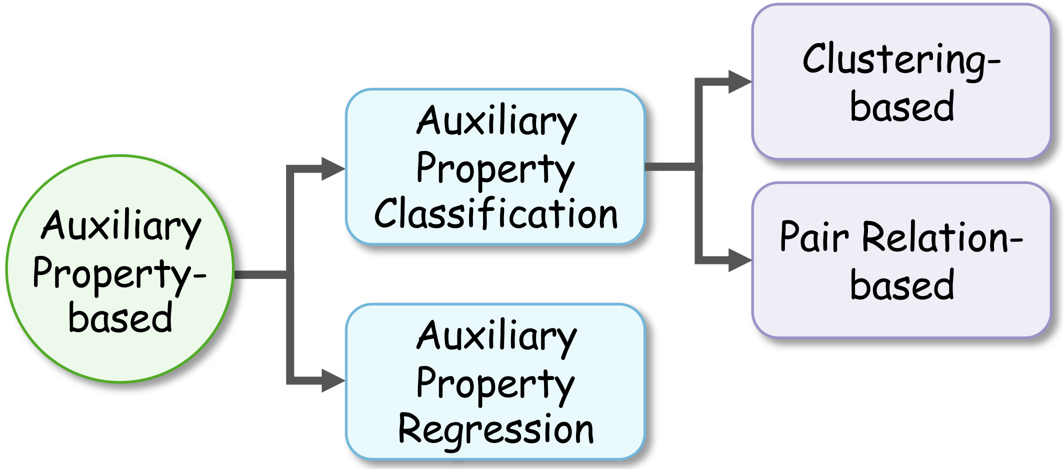
Auxiliary Property-based Methods: Origin

Generation-based methods aim to predict node-, link- and graph- level properties which can be obtained from the graph data freely.

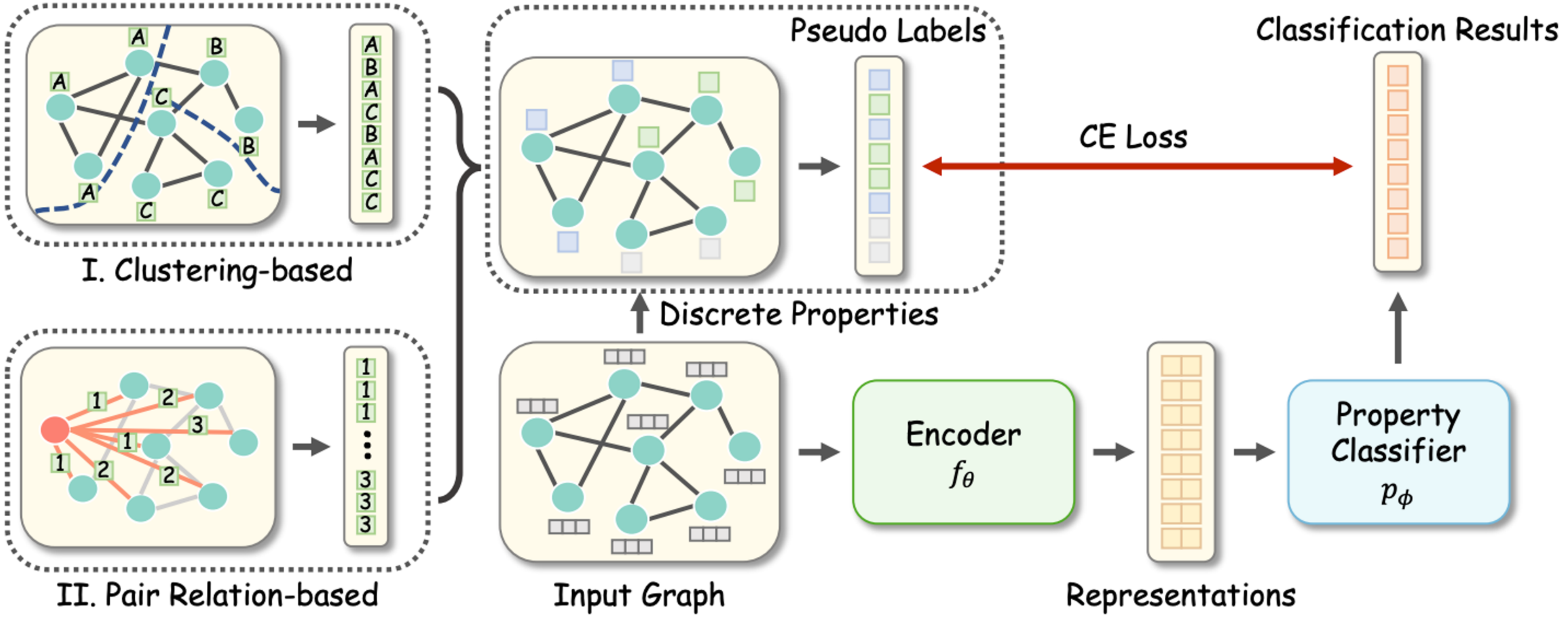
Origin: Supervised learning ⇒ Learn with "sample-label" pairs

Difference:
⇒ Supervised learning uses manual labels to train models
⇒ Auxiliary property-based methods uses pseudo labels to train models

Taxonomy: follows supervised learning



Auxiliary Property Classification



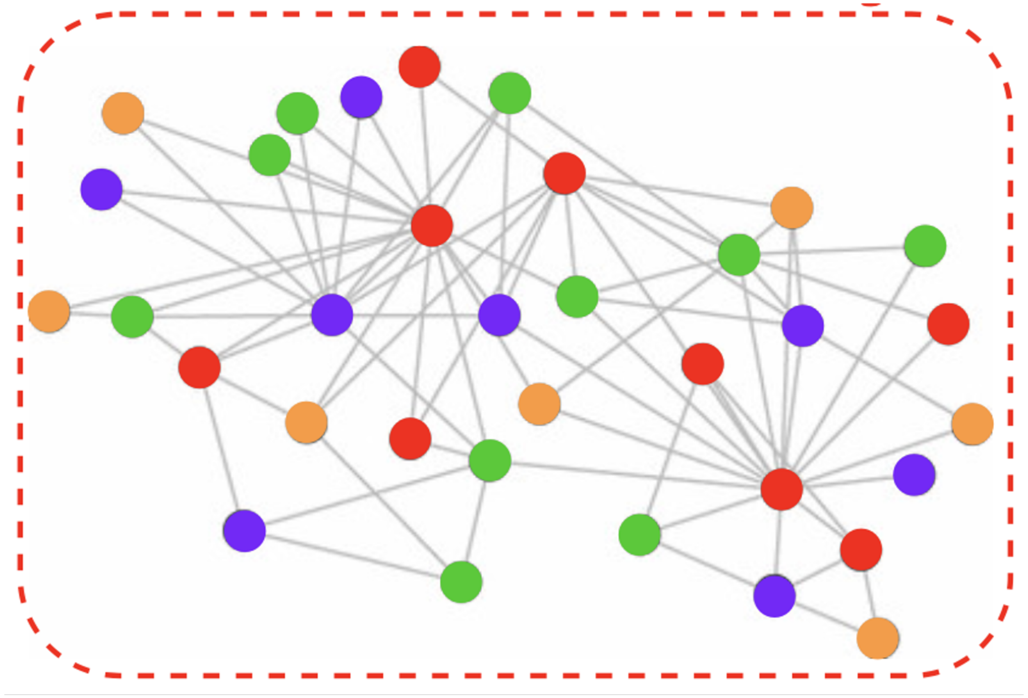
- **Pretext Decoder:** Classifier head
- **SSL Loss:** Classification Loss (Cross-entropy)

How to acquire properties?

- Clustering
- Pair Relation

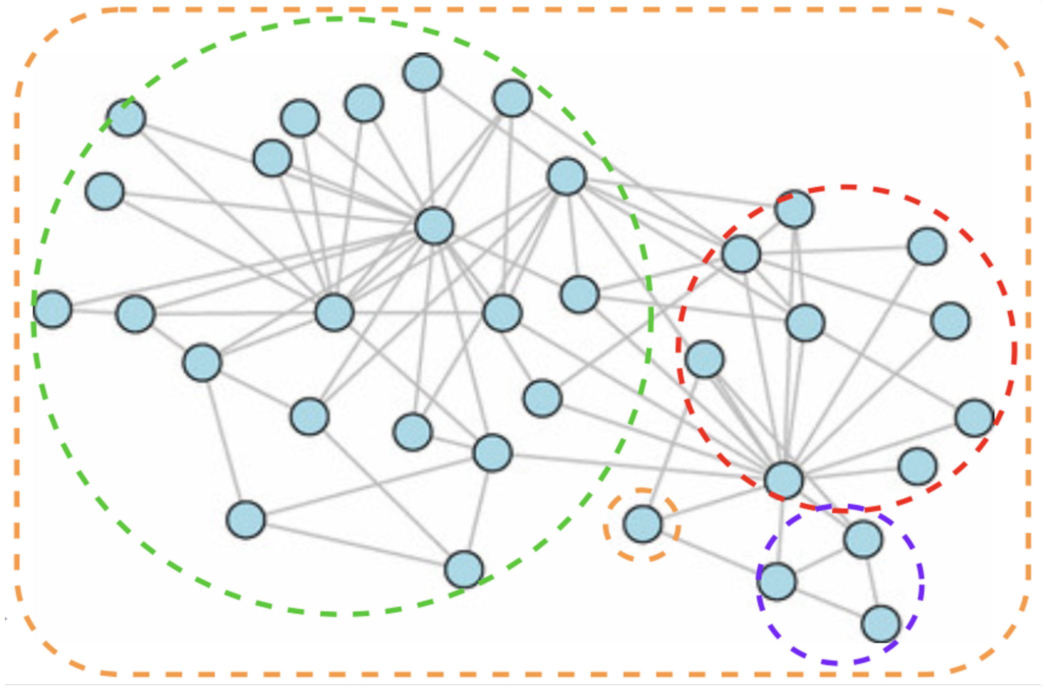
Clustering-based Auxiliary Property Classification: Representative Methods

- **Node Feature Clustering**



Feature-based clustering
(e.g., k-means)

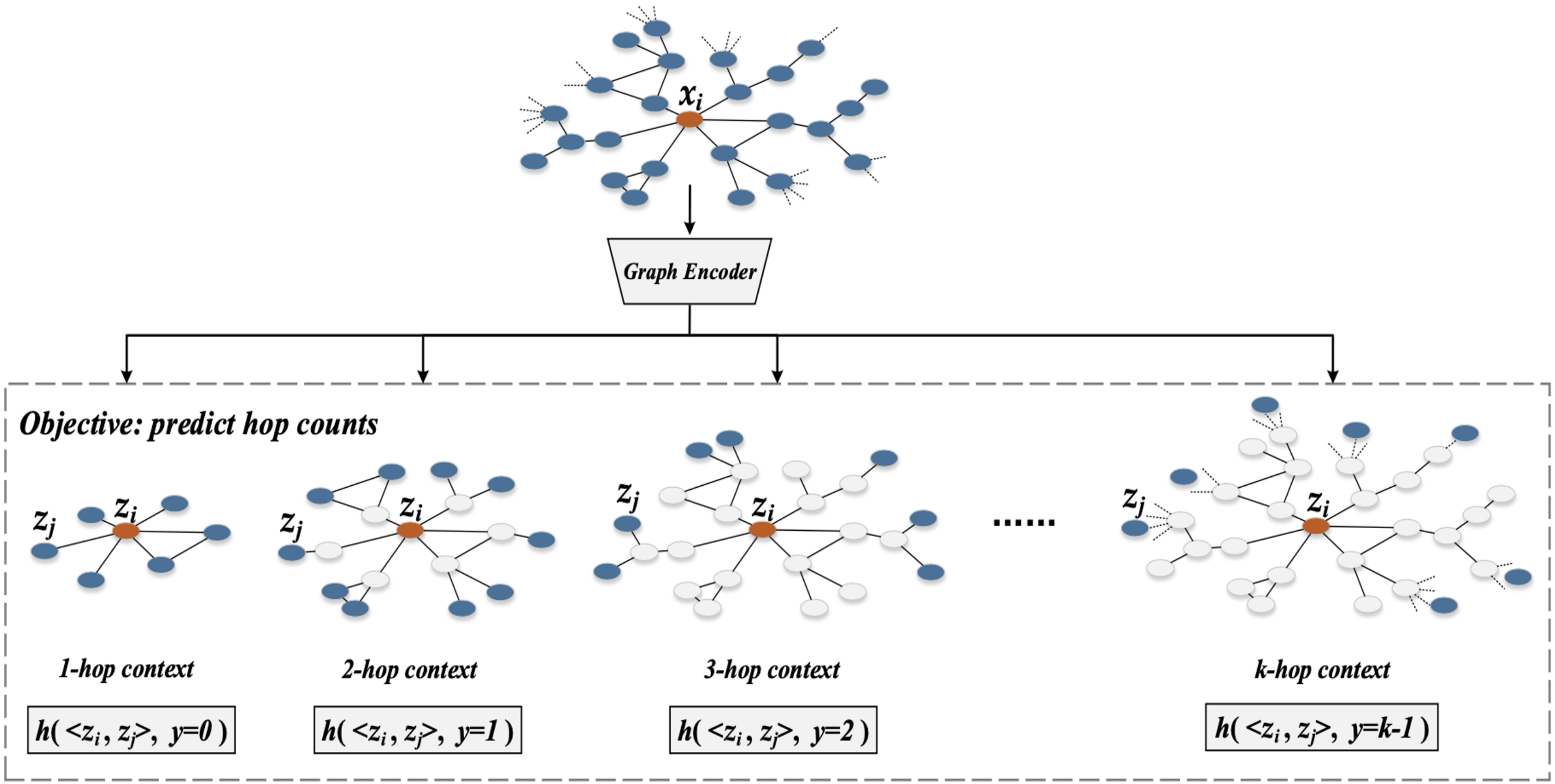
- **Graph Topology Partitioning**



Structure-based clustering
(e.g., Metis)

You, Y., Chen, T., Wang, Z., & Shen, Y. (2020, November). When does self-supervision help graph convolutional networks?. In International Conference on Machine Learning (pp. 10871-10880). PMLR.

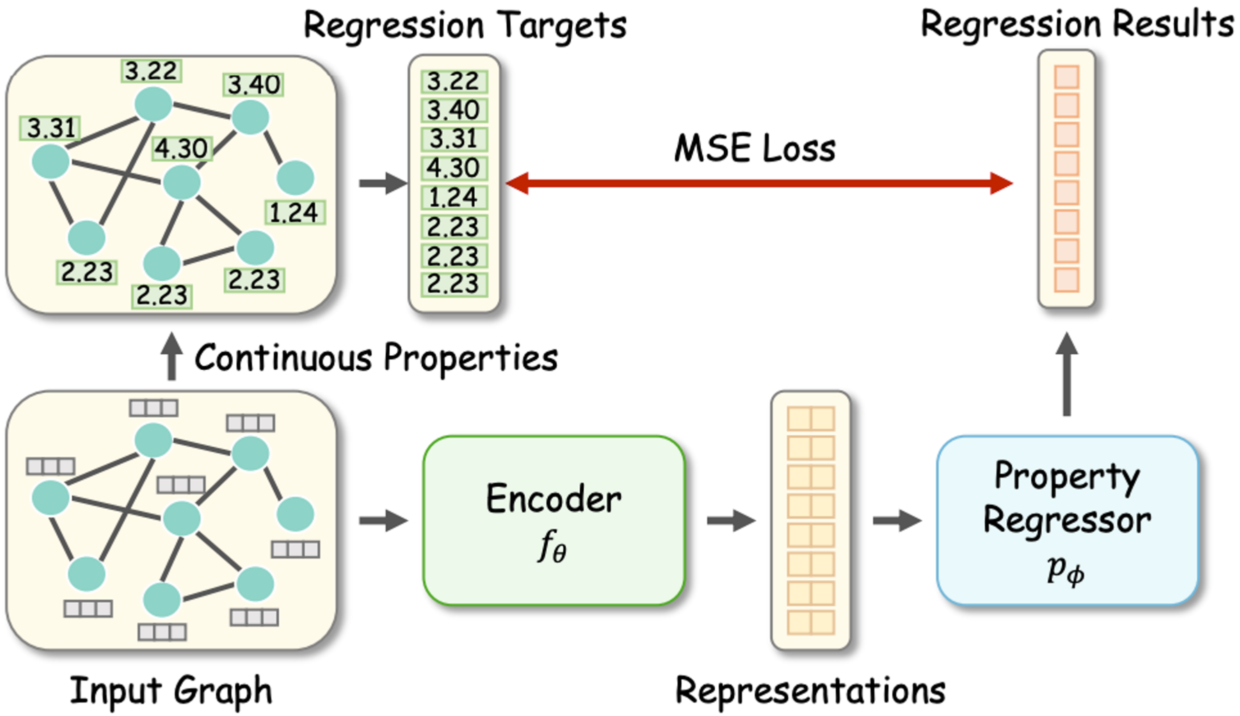
Pair Relation-based Auxiliary Property Classification: Representative Method



Peng, Z., Dong, Y., Luo, M., Wu, X. M., & Zheng, Q. (2020). Self-supervised graph representation learning via global context prediction. arXiv preprint arXiv:2003.01604.

Auxiliary Property Regression: Representative Method

- **NodeProperty**



- **Pretext Decoder:** Regression head
- **SSL Loss:** Regression Loss (MSE)

E.g., target property ⇒ the degree of nodes

Jin, W., Derr, T., Liu, H., Wang, Y., Wang, S., Liu, Z., & Tang, J. (2020). Self-supervised learning on graphs: Deep insights and new direction. arXiv preprint arXiv:2006.10141.

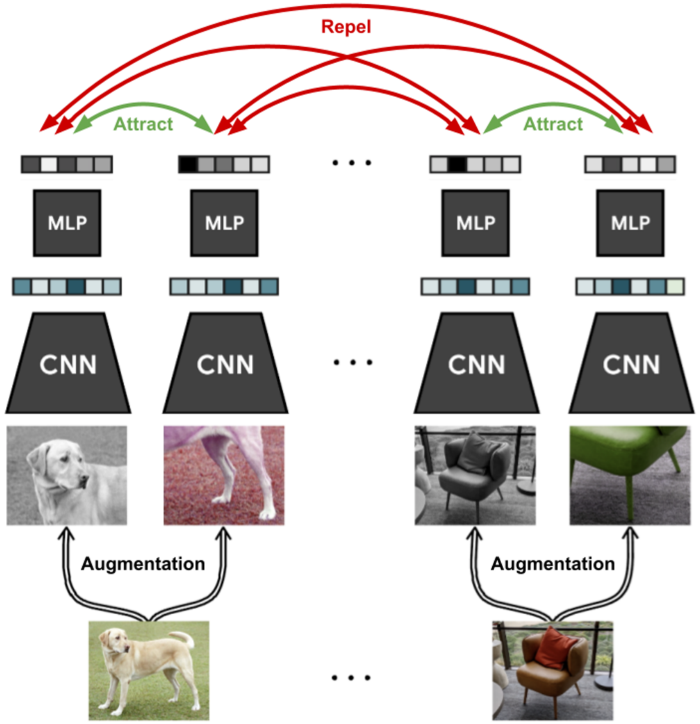
Auxiliary Property-based Methods: Summary

Approach	Pretext Task Category	Downstream Task Level	Training Scheme	Data Type of Graph	Property Level	Mapping Function
Node Clustering [17]	CAPC	Node	PF/JL	Attributed	Node	Feature-based Clustering
M3S [54]	CAPC	Node	JL	Attributed	Node	Feature-based Clustering
Graph Partitioning [17]	CAPC	Node	PF/JL	Attributed	Node	Structure-based Clustering
Cluster Preserving [48]	CAPC	Node/Link/Graph	PF	Attributed	Node	Structure-based Clustering
CAGNN [55]	CAPC	Node	URL	Attributed	Node	Feature-based Clustering with Structural Refinement
S^2GRL [56]	PAPC	Node/Link	URL	Attributed	Node Pair	Shortest Distance Function
PairwiseDistance [41]	PAPC	Node	PF/JL	Attributed	Node Pair	Shortest Distance Function
Centrality Score Ranking [48]	PAPC	Node/Link/Graph	PF	Attributed	Node Pair	Centrality Scores Comparison
TopoTER [57]	PAPC	Node/Graph	URL	Attributed	Node Pair	Topological Transformation Indicator
NodeProperty [41]	APR	Node	PF/JL	Attributed	Node	Degree Calculation
Distance2Cluster [41]	APR	Node	PF/JL	Attributed	Node Pair	Distance to Cluster Center
PairwiseAttrSim [41]	APR	Node	PF/JL	Attributed	Node Pair	Cosine Similarity of Feature
SimP-GCN [58]	APR	Node	JL	Attributed	Node Pair	Cosine Similarity of Feature

Contrast-based Methods: Origin

Contrast-based methods learn by maximizing the agreement between two augmented instances.

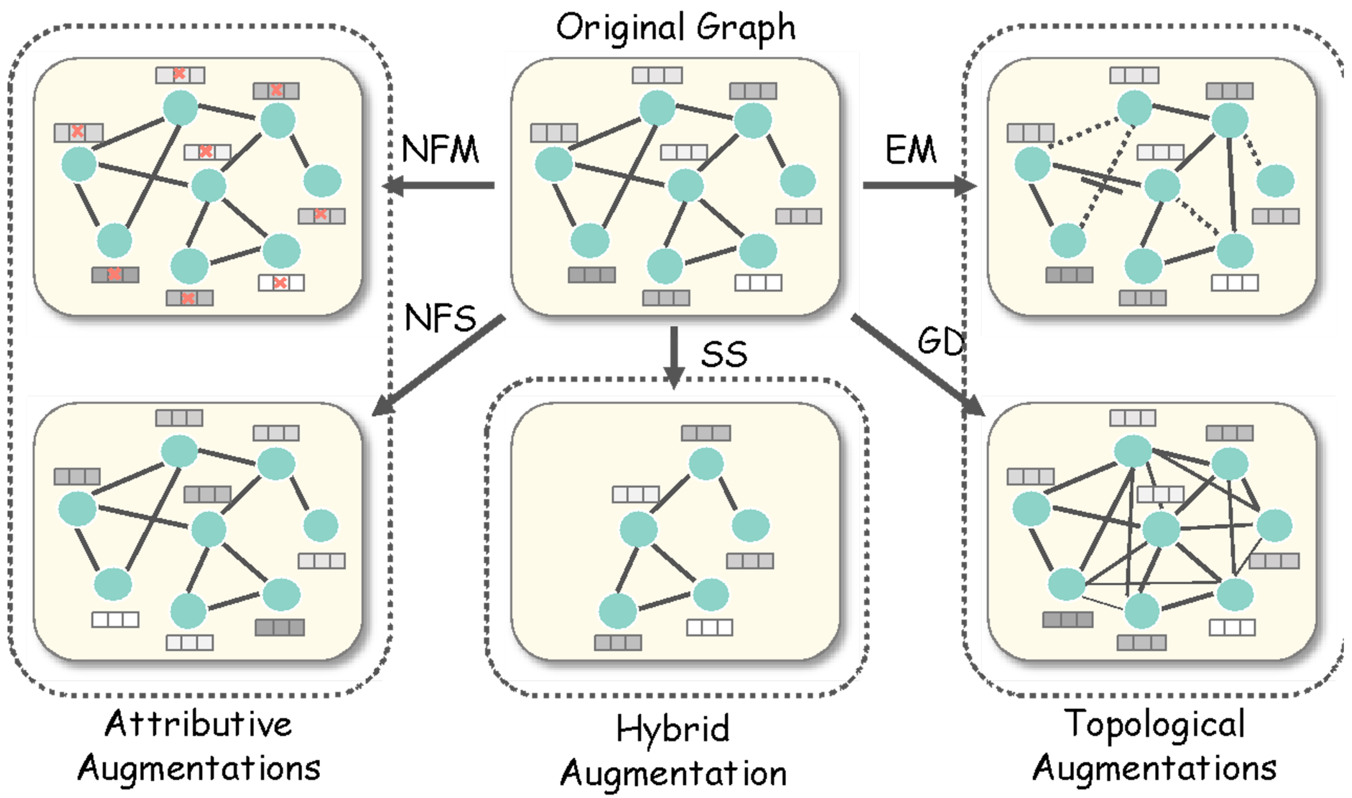
Origin: Visual Contrastive Learning ⇒ Mutual Information (MI) Maximization



Key components:

- Data augmentation
- Contrastive model <main taxonomy>
- Contrastive objective

Data Augmentation on Graphs

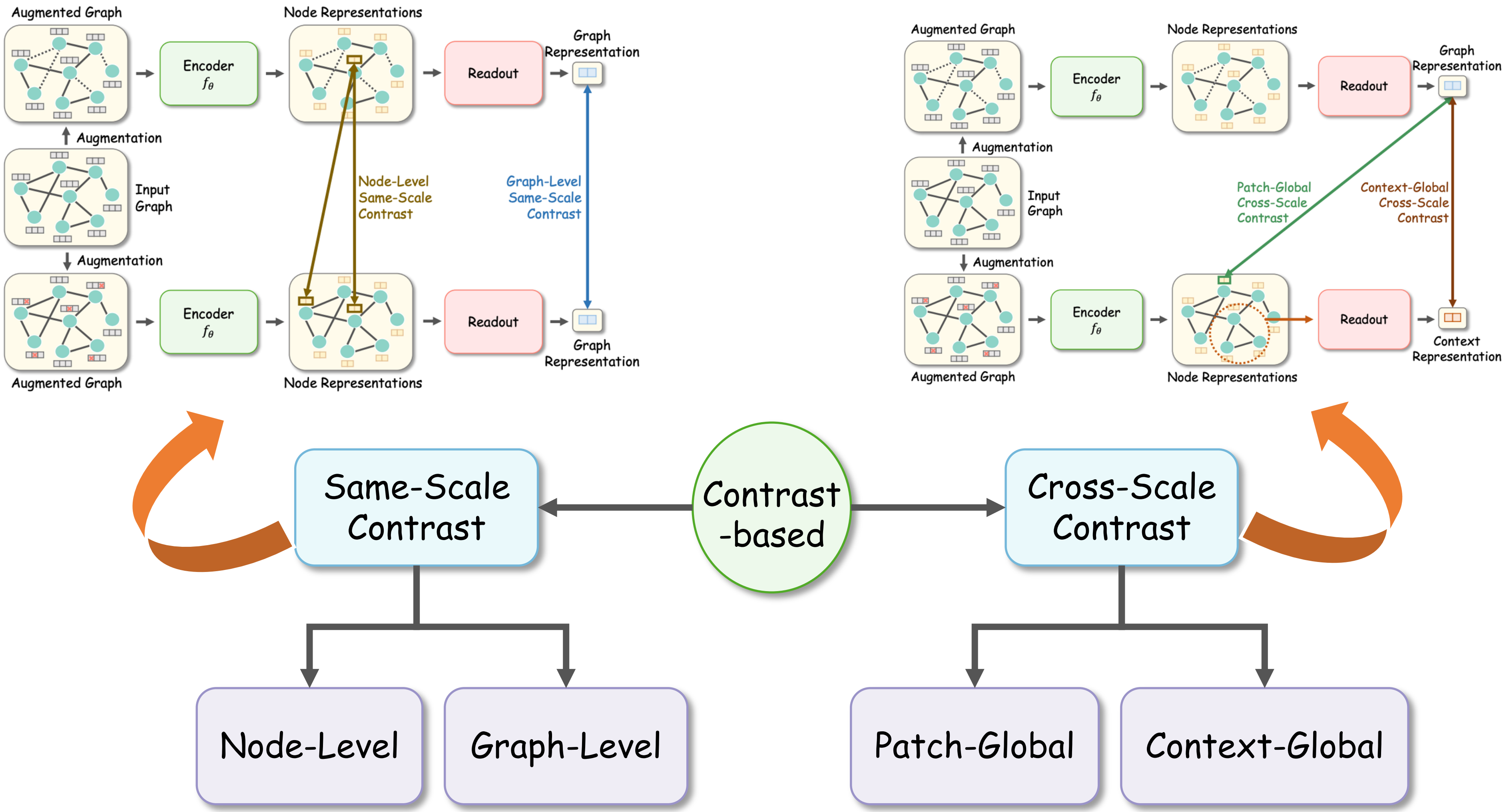


- Attributive augmentations
  - Node feature masking (NFM)
  - Node feature shuffle (NFS)

- Topological augmentations
  - Edge modification (EM)
  - Graph diffusion (GD)

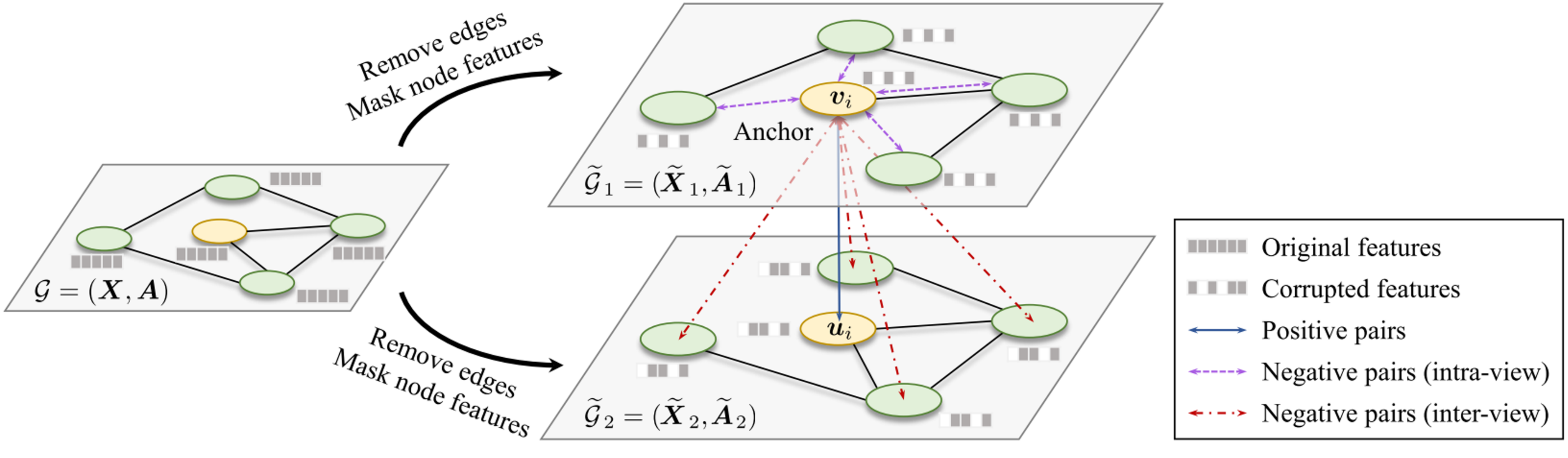
- Hybrid augmentations
  - Subgraph sampling (SS)

Graph Contrastive Learning: Taxonomy



Node-Level Same-Scale Contrast: Representative Method

- GRACE



- **SimCLR Contrastive Learning Framework**
- Intra + Inter view contrast
- Augmentation: Remove edges (EM) + mask features (NFM)

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2020). Deep graph contrastive representation learning. ICML Workshop

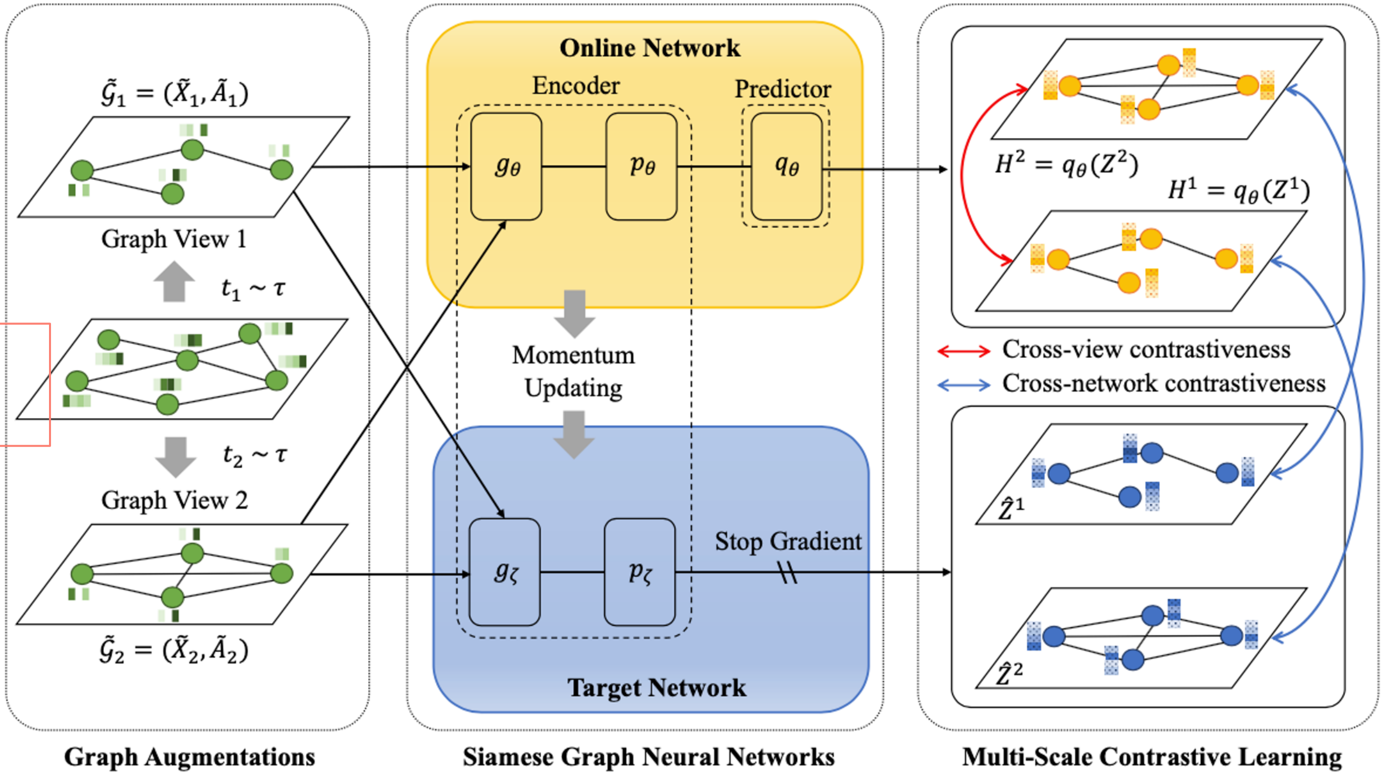
Node-Level Same-Scale Contrast: Representative Method

- **MERIT**

g_θ and g_ζ are two graph encoders

p_θ, p_ζ and q_θ are two-layer MLPs with the batch normalization

$t_1{\sim}\tau$ and $t_2{\sim}\tau$ are two different graph augmentations



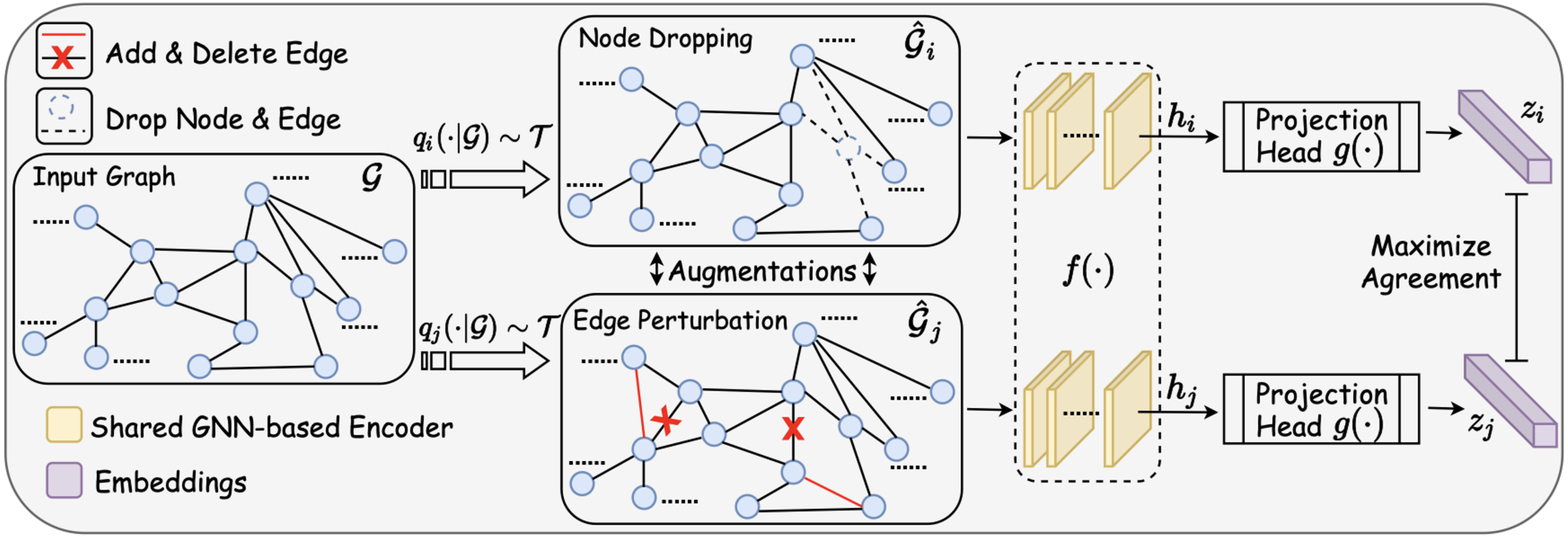
- Two graph views are first generated via graph augmentations
- Then, online and target networks are employed to generate node representations for each view

- A multi-scale graph contrastive schema with the self-knowledge distillation is proposed to train the online graph encoder

Jin, M., Zheng, Y., Li, Y. F., Gong, C., Zhou, C., & Pan, S. (2021). Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *International Joint Conference on Artificial Intelligence 2021* (pp. 1477-1483).

Graph-Level Same-Scale Contrast: Representative Method

- **GraphCL**

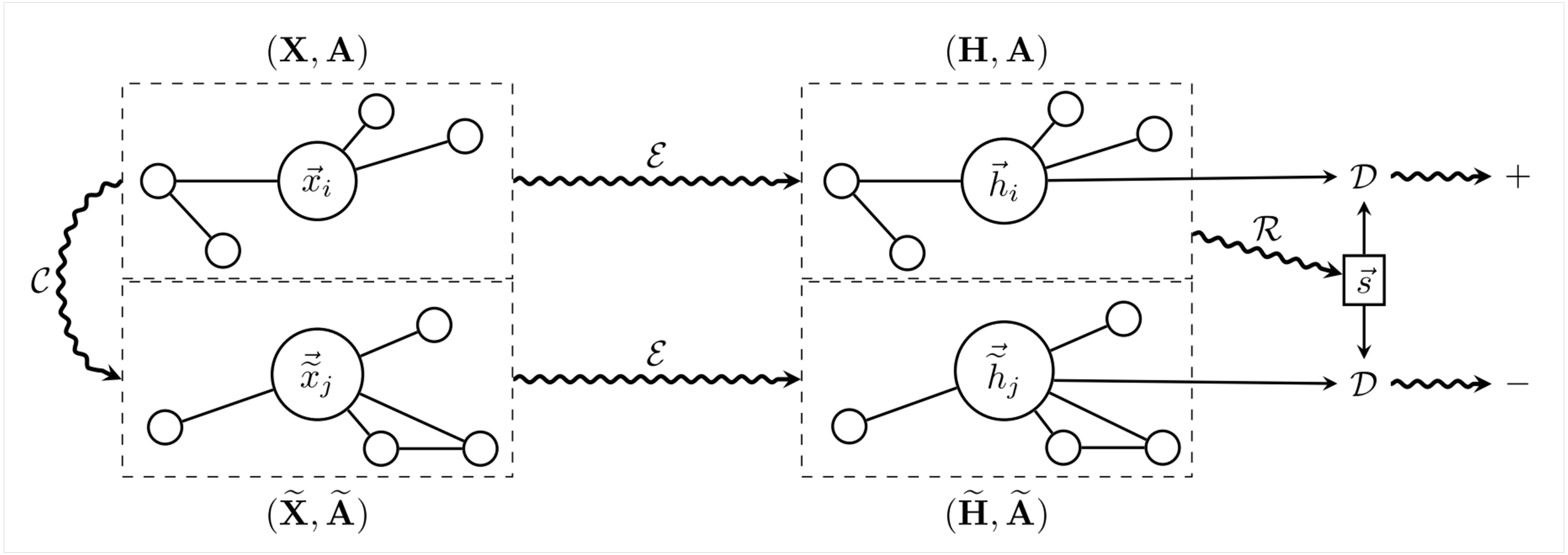


- **SimCLR Contrastive Learning Framework**
- Augmentation: EM+SS

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems, 33, 5812-5823.

Patch-Global Cross-Scale Contrast: Representative Method

- **DGI**

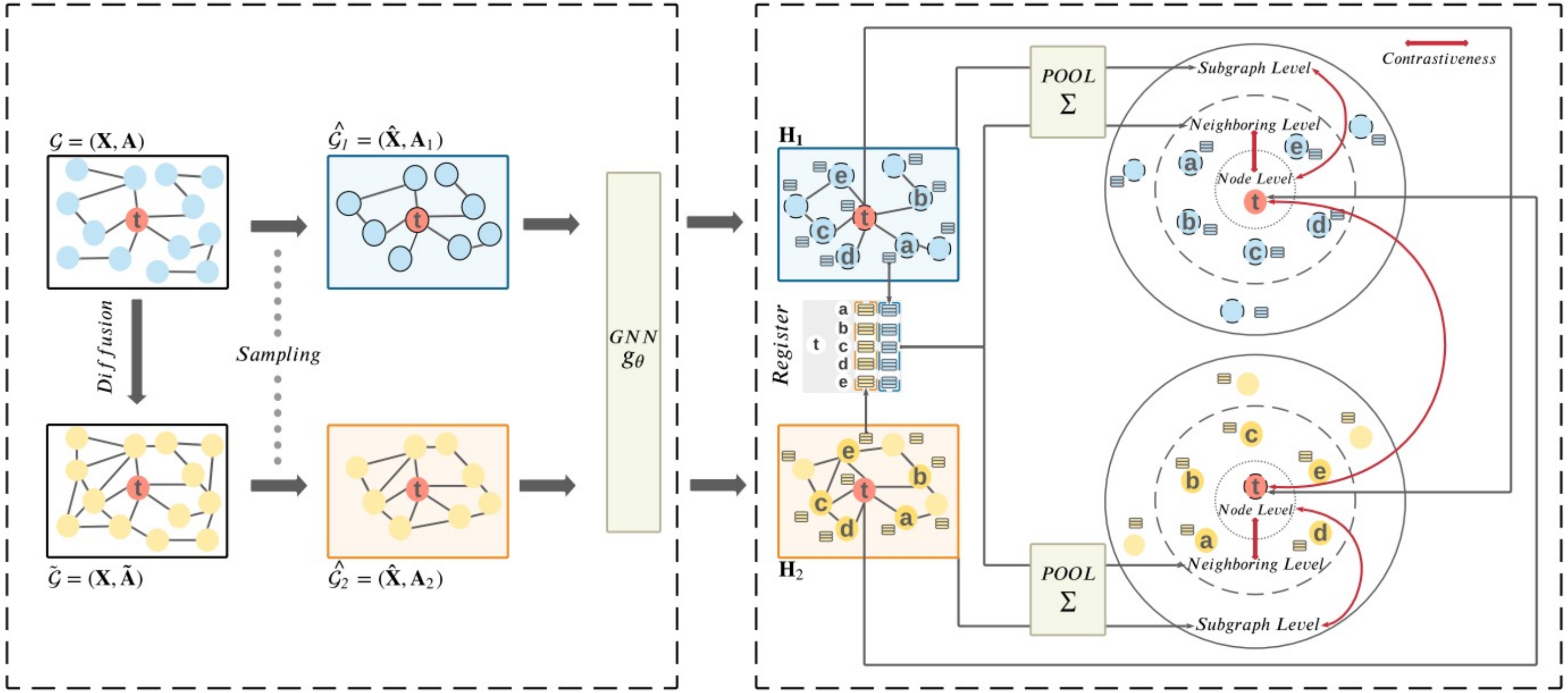


- **Maximize the MI between node and full graph**

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep Graph Infomax. ICLR (Poster), 2(3), 4.

Patch-Global Cross-Scale Contrast: Representative Method

- ***G-Zoom***



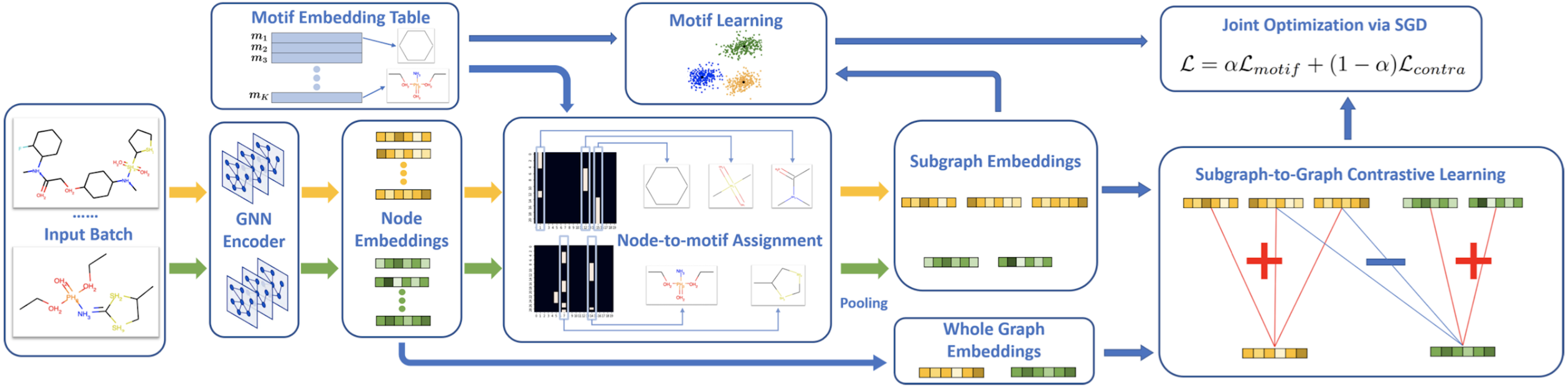
(a) Graph Augmentation and Encoding (b) Contrastive Learning with Adjusted Zooming

- Node vs. Node
- Node vs. Context
- Node vs. Graph

Zheng, Y., Jin, M., Pan, S., Li, Y. F., Peng, H., Li, M., & Li, Z. (2022). Toward Graph Self-Supervised Learning With Contrastive Adjusted Zooming. *IEEE Transactions on Neural Networks and Learning Systems.*

Context-Global Cross-Scale Contrast: Representative Method

- ***MICRO-Graph***



- **Motif vs. Full graph**

Subramonian, A. (2021, May). MOTIF-Driven Contrastive Learning of Graph Representations. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 18, pp. 15980-15981).

MI Estimation - Contrastive Loss

- Jensen-Shannon Estimator

$$\mathcal{MI}_{JSD}(\mathbf{h}_i, \mathbf{h}_j) = \mathbb{E}_{\mathcal{P}}\Big[\log\big(\mathcal{D}(\mathbf{h}_i, \mathbf{h}_j)\big)\Big] - \mathbb{E}_{\mathcal{P}\times\widetilde{\mathcal{P}}}\Big[\log\big(1 - \mathcal{D}(\mathbf{h}_i, \mathbf{h'}_j)\big)\Big].$$

- Noise-Contrastive Estimator

$$\mathcal{MI}_{NCE}(\mathbf{h}_i, \mathbf{h}_j) = \mathbb{E}_{\mathcal{P}\times\widetilde{\mathcal{P}}^N}\left[\log \frac{e^{\mathcal{D}(\mathbf{h}_i,\mathbf{h}_j)}}{e^{\mathcal{D}(\mathbf{h}_i,\mathbf{h}_j)} + \sum_{n\in N} e^{\mathcal{D}(\mathbf{h}_i,\mathbf{h}'_n)}}\right]$$

- Triplet loss

$$\mathcal{L}_{triplet} = \mathbb{E}_{\mathcal{P}\times\widetilde{\mathcal{P}}}\Big[\max\big[\mathcal{D}(\mathbf{h}_i, \mathbf{h}_j) - \mathcal{D}(\mathbf{h}_i, \mathbf{h'}_j) + \epsilon, 0\big]\Big]$$

- BYOL loss

$$\mathcal{L}_{byol} = \mathbb{E}_{\mathcal{P}^N}\left[-\frac{2}{N}\sum_{i,j\in N}\frac{\left[p_\psi(\mathbf{h}_i)\right]^T\mathbf{h}_j}{\|p_\psi(\mathbf{h}_i)\|\,\|\mathbf{h}_j\|}\right]$$

- Barlow Twins loss

$$\mathcal{L}_{bt} = \mathbb{E}_{\mathbf{B}\sim\mathcal{P}^N}\left[\sum_a\Big(1 - \frac{\sum_{i\in\mathbf{B}}\mathbf{H}^{(1)}_{ia}\mathbf{H}^{(2)}_{ia}}{\left\|\mathbf{H}^{(1)}_{ia}\right\|\left\|\mathbf{H}^{(2)}_{ia}\right\|}\Big)^2 + \lambda\sum_a\sum_{b\neq a}\left(\frac{\sum_{i\in\mathbf{B}}\mathbf{H}^{(1)}_{ia}\mathbf{H}^{(2)}_{ib}}{\left\|\mathbf{H}^{(1)}_{ia}\right\|\left\|\mathbf{H}^{(2)}_{ib}\right\|}\right)^2\right]$$

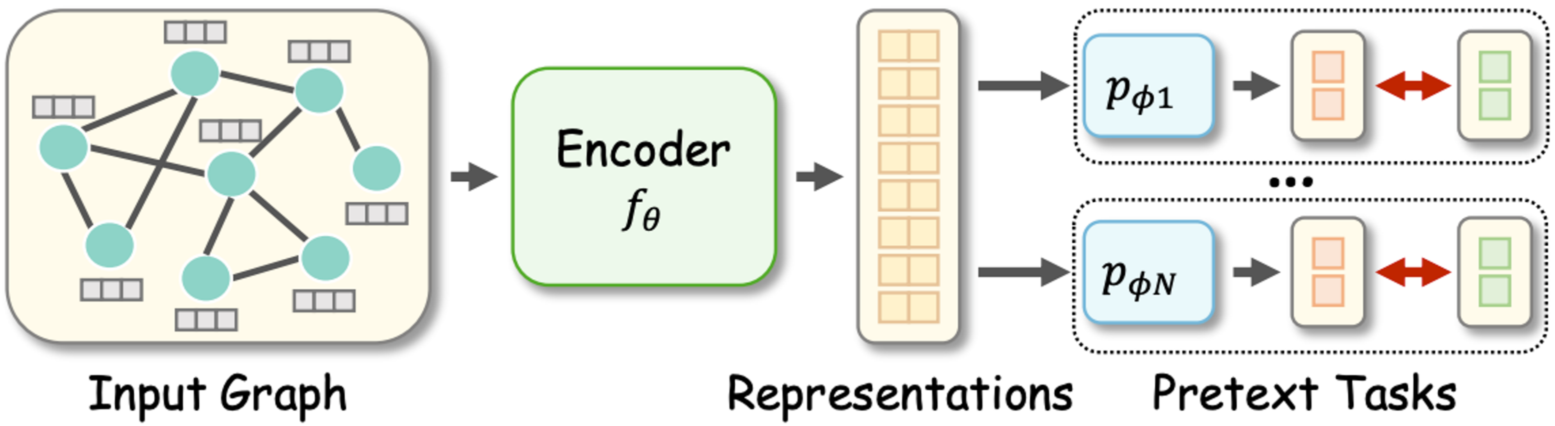
Contrast-based Methods: Summary

Approach	Pretext Task Category	Downstream Task Level	Training Scheme	Data Type of Graph	Graph Augmentation	Objective Function
DeepWalk [30]	NSC	Node	URL	Plain	SS	SkipGram
node2vec [31]	NSC	Node	URL	Plain	SS	SkipGram
GraphSAGE [78]	NSC	Node	URL	Attributed	SS	JSD
SELAR [80]	NSC	Node	JL	Heterogeneous	Meta-path sampling	JSD
LINE [79]	NSC	Node	URL	Plain	SS	JSD
GRACE [33]	NSC	Node	URL	Attributed	NFM+EM	InfoNCE
GROC [18]	NSC	Node	URL	Attributed	NFM+Adversarial EM	InfoNCE
GCA [67]	NSC	Node	URL	Attributed	Adaptive NFM+Adaptive EM	InfoNCE
GraphCL(N) [81]	NSC	Node	URL	Attributed	SS+NFS+EM	InfoNCE
GCC [15]	NSC	Node/Graph	PF/URL	Plain	SS	InfoNCE
HeCo [82]	NSC	Node	URL	Heterogeneous	NFM	InfoNCE
Contrast-Reg [71]	NSC	Node	JL	Attributed	Arbitrary	JSD
BGRL [83]	NSC	Node	URL	Attributed	NFM+EM	BYOL
SelfGNN [84]	NSC	Node	URL	Attributed	GD+Node attributive transformation	BYOL
G-BT [86]	NSC	Node	URL	Attributed	NFM+EM	Barlow Twins
MERIT [66]	NSC	Node	URL	Attributed	SS+GD+NFM+EM	BYOL+InfoNCE
DwGCL [68]	NSC	Node	JL	Attributed	Adaptive NFM+Adaptive EM	KL-Divergence
GraphCL(G) [65]	GSC	Graph	PF/URL	Attributed	SS+NFM+EM	InfoNCE
DACL [88]	GSC	Graph	URL	Attributed	Noise Mixing	InfoNCE
AD-GCL [75]	GSC	Graph	PF/URL	Attributed	Adversarail EM	InfoNCE
JOAO [69]	GSC	Graph	PF/URL	Attributed	Automated	InfoNCE
CSSL [74]	GSC	Graph	PF/JL/URL	Attributed	SS+Node insertion/deletion+EM	InfoNCE
LCGNN [89]	GSC	Graph	JL	Attributed	Arbitrary	InfoNCE
IGSD [74]	GSC	Graph	JL/URL	Attributed	GD+EM	BYOL+InfoNCE
DGI [13]	PGCC	Node	URL	Attributed	None	JSD
GIC [90]	PGCC	Node	URL	Attributed	Arbitrary	JSD
HDGI [91]	PGCC	Node	URL	Heterogeneous	None	JSD
ConCH [92]	PGCC	Node	JL	Attributed	None	JSD
DMGI [93]	PGCC	Node	JL/URL	Heterogeneous	None	JSD
EGI [94]	PGCC	Node	PF/JL	Attributed	SS	JSD
STDGI [70]	PGCC	Node	URL	Dynamic	Node feature shuffling	JSD
KS2L [95]	PGCC	Node	URL	Attributed	None	InfoNCE
MVGRL [14]	PGCC	Node/Graph	URL	Attributed	GD+SS	JSD
SUBG-CON [77]	PGCC	Node	URL	Attributed	SS+Node representation shuffling	Triplet
SLiCE [96]	PGCC	Edge	JL	Heterogeneous	None	JSD
InfoGraph [97]	PGCC	Graph	JL/URL	Attributed	None	JSD
Robinson et al. [98]	PGCC	Graph	URL	Attributed	Arbitrary	JSD
BiGI [99]	CGCC	Graph	URL	Heterogeneous	SS	JSD
HTC [100]	CGCC	Graph	JL	Attributed	NFS	JSD
MICRO-Graph [101]	CGCC	Graph	URL	Attributed	SS	InfoNCE
SUGAR [102]	CGCC	Graph	JL	Attributed	SS	JSD

Hybrid Methods: Motivation

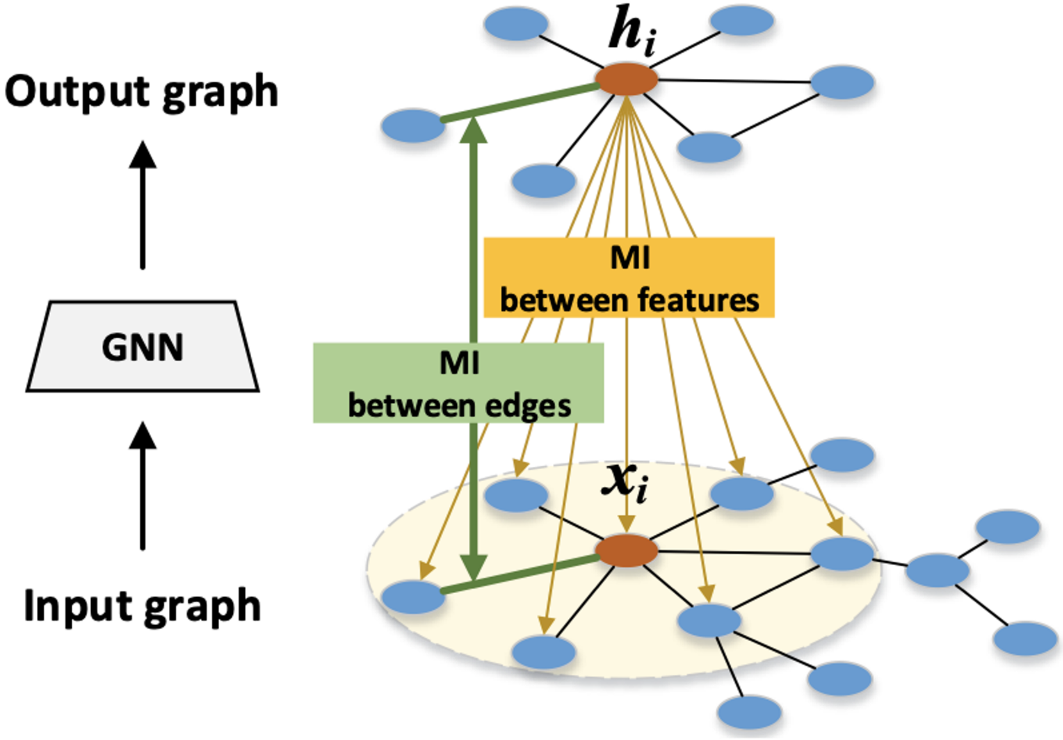
Hybrid methods integrate various pretext tasks together in a multi-task learning fashion

Motivation:
⇒ A single pretext task cannot provide sufficient guidance
⇒ Using multiple pretext tasks can better leverage the advantages of various types of supervision signals



Hybrid Methods: Representative Methods

- GMI

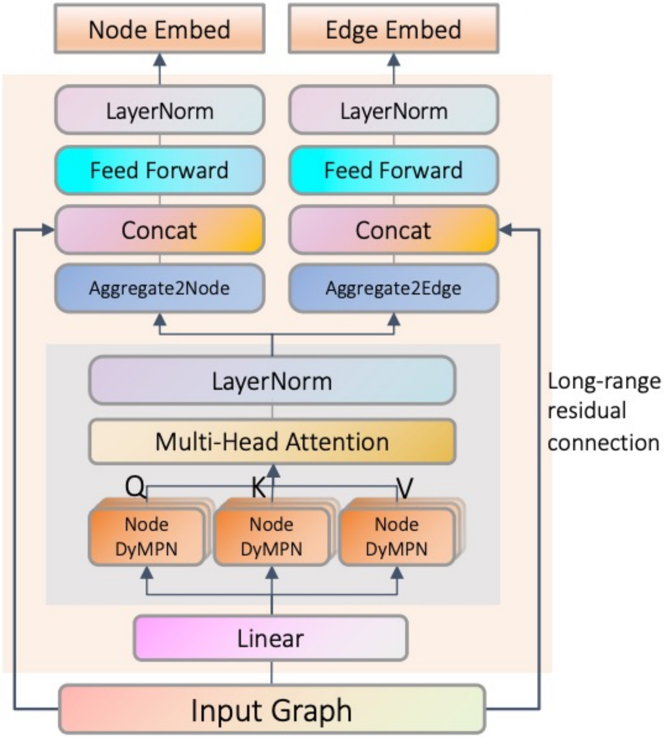


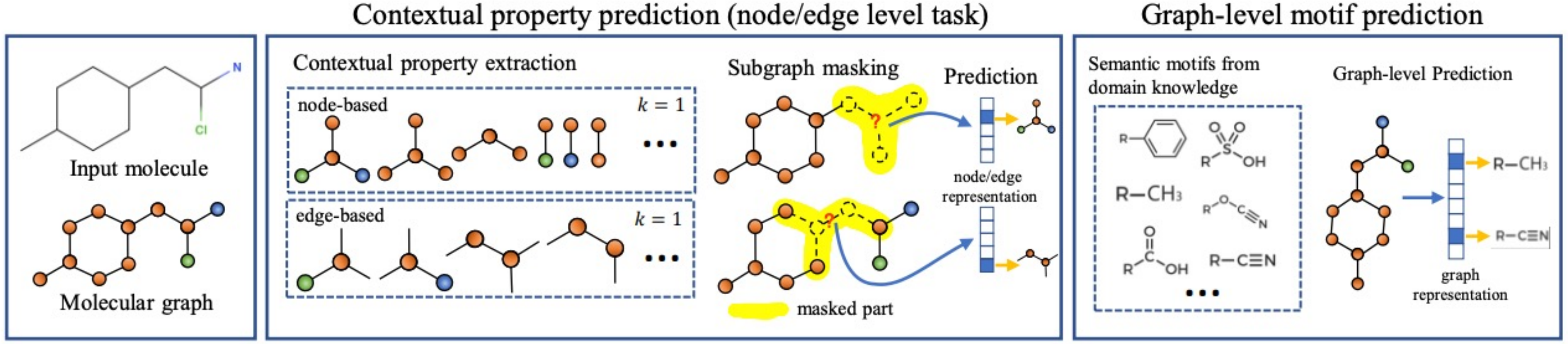
- Edge MI: Structure generation
- Node MI: Same-scale contrast

Peng, Z., Huang, W., Luo, M., Zheng, Q., Rong, Y., Xu, T., & Huang, J. (2020, April). Graph representation learning via graphical mutual information maximization. In Proceedings of The Web Conference 2020 (pp. 259-270).

Hybrid Methods: Representative Methods

- GROVER





- Node- and edge-level reconstruction

- Context- and graph-level auxiliary properties prediction

- Backbone model: Node and edge GNN transformers

Hu, Z., Dong, Y., Wang, K., Chang, K. W., & Sun, Y. (2020, August). Gpt-gnn: Generative pre-training of graph neural networks. In Proceedings of the 26th ACM SIGKDD (pp. 1857-1867).

Hybrid Methods: Summary

Approach	Pretext Task Categories	Downstream Task Level	Training Scheme	Data Type of Graph
GPT-GNN [9]	FG/SG	Node/Link	PF	Hetero.
Graph-Bert [104]	FG/SG	Node	PF	Attributed
PT-DGNN [105]	FG/SG	Link	PF	Dynamic
M. et al. [43]	FG/FG/FG	Node	JL	Attributed
GMI [106]	SG/NSC	Node/Link	URL	Attributed
CG^3 [107]	SG/NSC	Node	JL	Attributed
MVMI-FT [108]	SG/PGCC	Node	URL	Attributed
GraphLoG [109]	NSC/GSC/ CGCC	Graph	PF	Attributed
HDMI [110]	NSC/PGCC	Node	URL	Multiplex
LnL-GNN [111]	NSC/NSC	Node	JL	Attributed
Hu et al. [48]	SG/APC/ APC	Node/Link/ Graph	PF	Attributed
GROVER [10]	APC/APC	Node/Link/ Graph	PF	Attributed
Kou et al. [112]	FG/SG/ APC	Node	JL	Attributed

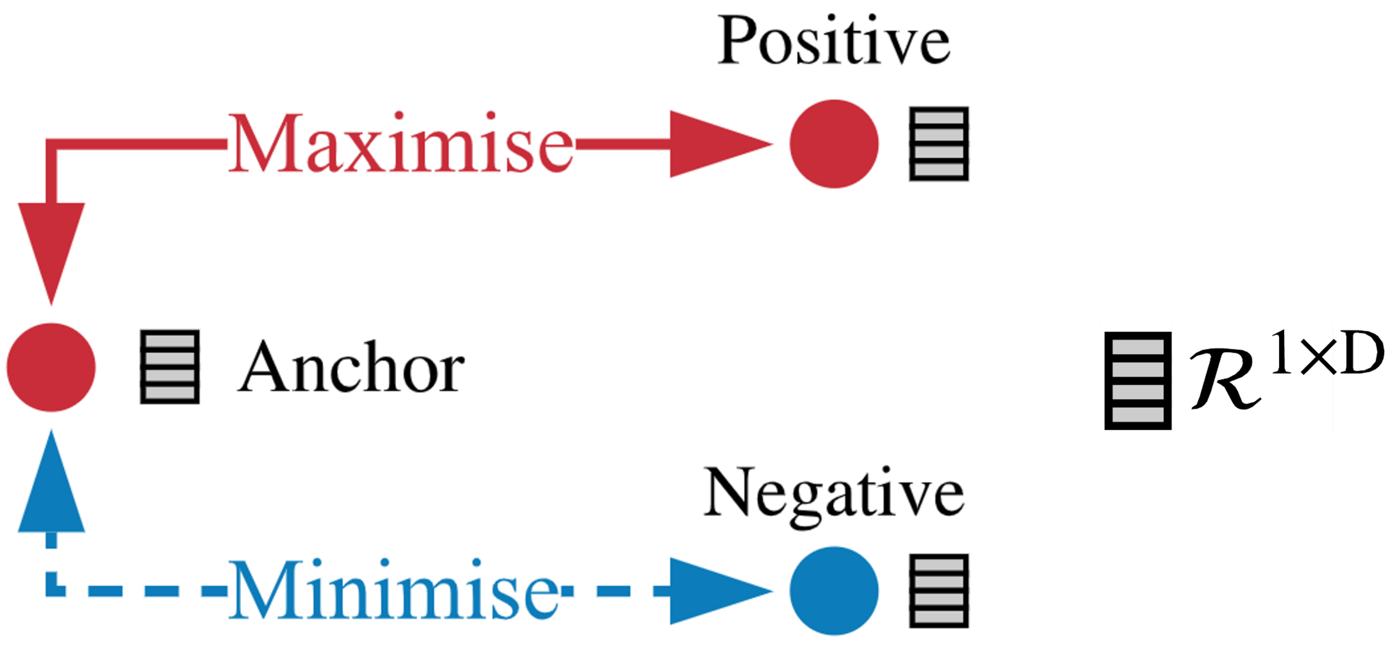
Part 3: Frontiers of graph self-supervised learning

- Efficient graph self-supervised learning : A new paradigm
- Heterophilic graph self-supervised learning
- Heterogeneous graph self-supervised learning

Efficient graph self-supervised learning : A new paradigm

Existing Problems - Slow Computation with Node Comparison

These contrastive-learning approaches rely on node-to-node comparison.



(a)Node-to-node Comparison

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Existing Problems - Slow Computation with Node Comparison

Node-to-node comparison require heavy gradient computation. For example, for the two representative contrastive losses:

InfoNCE Loss

$$\mathcal{L}_{\mathrm{NCE}}(\mathrm{i}) = -\log \frac{e^{z_i \cdot c_i/\tau}}{\boxed{\sum_{k=1}^{N} e^{z_i \cdot z_k/\tau}}},$$

Gradient Computation require all negative samples

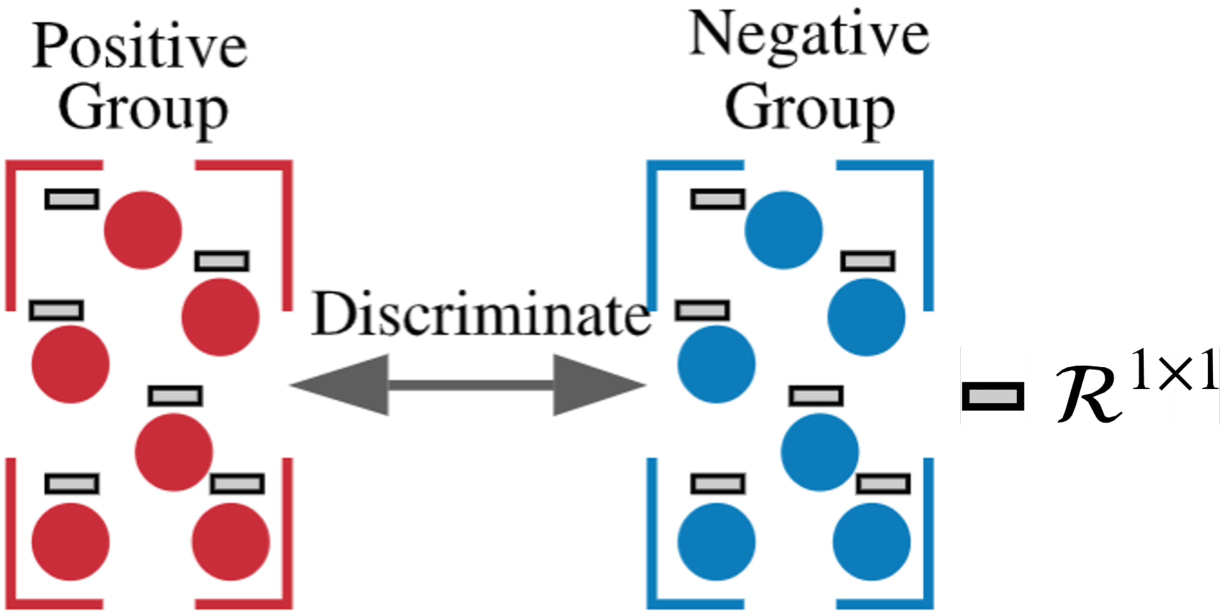
JSD-estimator

$$\mathcal{L}_{\mathrm{JSD}}(\mathrm{i}) = -\log \mathcal{D}(z_i, \boxed{\vec{s}}) + \log(1 - \mathcal{D}(\tilde{z}_i, \vec{s})), \quad \vec{s} = \sigma(\frac{1}{N}\sum_{i=1}^{N} z_i)$$

Gradient Computation require all positive samples

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Introduction to Group Discrimination (GD)



(b) Group Discrimination

Summarisation (e.g., sum):

$\mathcal{R}^{1\times D}$ ⇨ $\mathcal{R}^{1\times 1}$

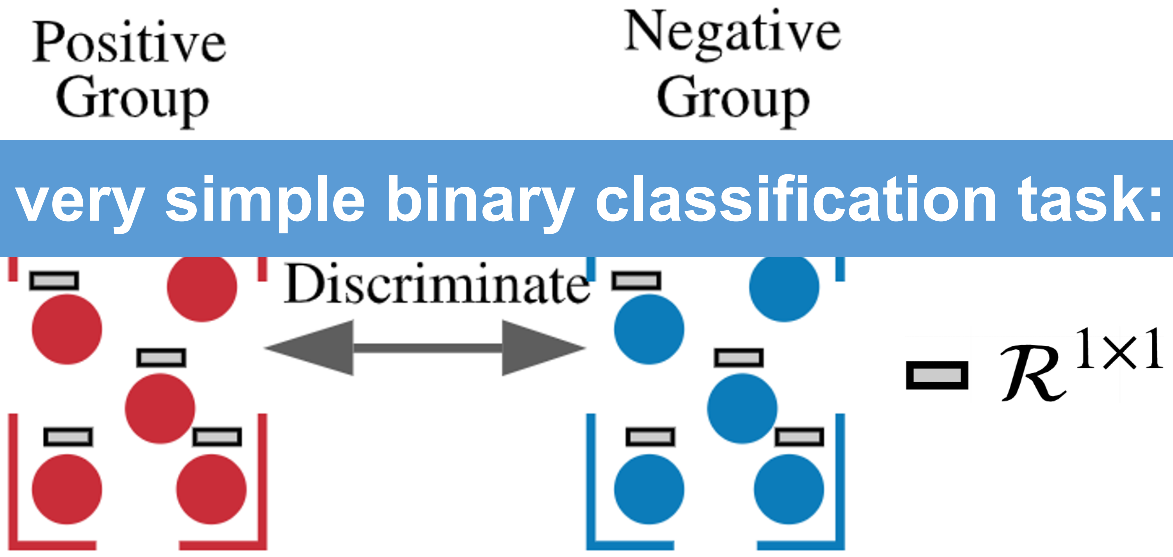
Positive Group:
Summarised Node representation$\mathcal{R}^{1\times1}$)
generated with original or augmented grap

Negative Group:
Summarised Node representation$\mathcal{R}^{1\times1}$)
generated with corrupted graph.

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Introduction to Group Discrimination (GD)

Use a very simple BCE loss to conduct discrimination



(b) Group Discrimination

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{2N}\left(\sum_{i=1}^{2N} \mathrm{y_i} \log \mathrm{h_i} + (1-\mathrm{y_i}) \log(1-\mathrm{h_i})\right)$$

A very simple binary classification task: discriminating positive/negative samples

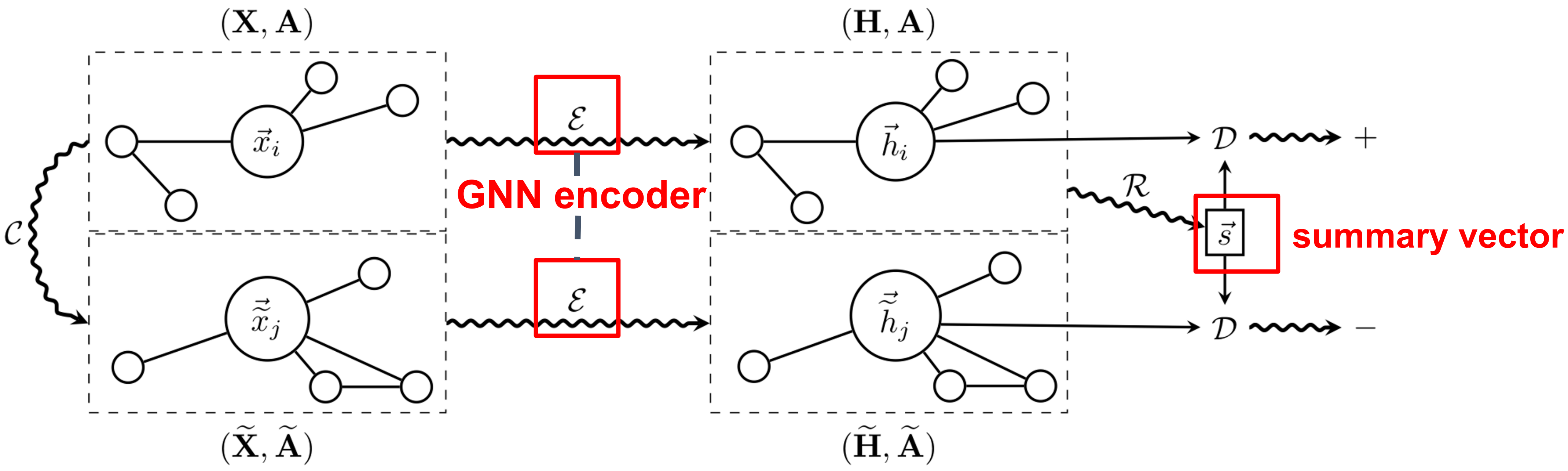
If positive → y = 1, else → y = 0

$\mathrm{h_i} \in \mathcal{R}^{1\times1}$ is the summarised node embedding/binary prediction for a node i

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

Original Thought of DGI →
MI maximization between nodes and summary vector.



$$\mathcal{L}_{\mathrm{DGI}} = \frac{1}{2\mathrm{N}}\left(\sum_{\mathrm{i}=1}^{\mathrm{N}} \log \mathcal{D}(\mathrm{z_i}, \vec{\mathrm{s}}) + \log(1 - \mathcal{D}(\tilde{\mathrm{z}}_\mathrm{i}, \vec{\mathrm{s}}))\right),$$

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

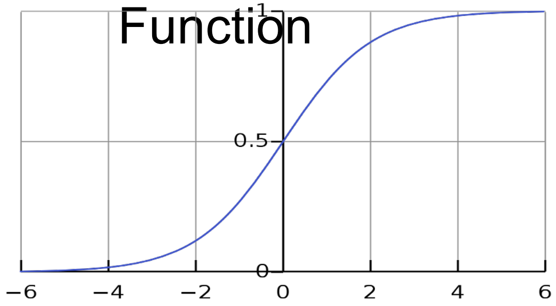
However, due to inappropriate usage of Sigmoid function….

$$\mathcal{L}_{\mathrm{DGI}} = \frac{1}{2N}\left(\sum_{i=1}^{N} \log \mathcal{D}(z_i, \vec{s}) + \log(1 - \mathcal{D}(\tilde{z}_i, \vec{s}))\right),$$

$$\vec{s} = \boxed{\sigma}\left(\frac{1}{N}\sum_{i=1}^{N} \boxed{z_i}\right)$$

Sigmoid Function

Node Embeddings



Rethinking DGI

Activation	Statistics	Cora	CiteSeer	PubMed
ReLU/LReLU/PReLU	Mean	0.50	0.50	0.50
	Std	1.3e-03	1.0e-04	4.0e-04
	Range	1.4e-03	8.0e-04	1.5e-03
Sigmoid	Mean	0.62	0.62	0.62
	Std	5.4e-05	2.9e-05	6.6e-05
	Range	3.6e-03	3.0e-03	3.2e-03

Value in summary vector $\vec{s}$ almost becomes constant vector $s = \epsilon I = I$ with no variance.

The assumption of learning via MI interaction between nodes and summary vector

Dataset	0	0.2	0.4	0.6	0.8	1.0
Cora	70.3±0.7	82.4±0.2	82.3±0.3	82.5±0.4	82.3±0.3	82.5±0.1
CiteSeer	61.8±0.8	71.7±0.6	71.9±0.7	71.6±0.9	71.7±1.0	71.6±0.8
PubMed	68.3±1.5	77.8±0.5	77.9±0.8	77.7±0.9	77.4±1.1	77.2±0.9

Changing ϵ has trivial effect on model performance.

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

Simplifying DGI

Set ϵ to 1 for $s = \epsilon I = I$, and remove w in $\mathcal{D}(z_i, \vec{s}) = z_i \cdot w \cdot \vec{s},$

$$\mathcal{L}_{DGI} = \frac{1}{2N}\left(\sum_{i=1}^{N} \log \mathcal{D}(z_i, \vec{s}) + \log(1 - \mathcal{D}(\tilde{z}_i, \vec{s}))\right),$$

$$= \frac{1}{2N}\left(\sum_{i=1}^{N} \log(z_i \cdot \vec{s}) + \log(1 - \tilde{z}_i \cdot \vec{s})\right),$$

$$= \frac{1}{2N}\left(\sum_{i=1}^{N} \log(\mathrm{sum}(z_i)) + \log(1 - \mathrm{sum}(\tilde{z}_i))\right),$$

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

$$= \frac{1}{2N}(\sum_{i=1}^{N} \log(\mathrm{sum}(z_i)) + \log(1 - \mathrm{sum}(\tilde{z}_i))),$$

Considering summarised embedding as h_i → become BCE loss

$$\mathcal{L}_{BCE} = \frac{1}{2N}(\sum_{i=1}^{2N} y_i \log h_i + (1 - y_i)\log(1 - h_i)),$$

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

With the new loss → Dramatic improvement in memory and time

Experiment	Method	Cora	CiteSeer	PubMed
Accuracy	DGI	81.7±0.6	71.5±0.7	77.3±0.6
	DGI_{BCE}	82.5±0.3	71.7±0.6	77.7±0.5
Memory	DGI	4189MB	8199MB	11471MB
	DGI_{BCE}	1475MB|64.8%	1587MB|80.6%	1629MB|85.8%
Time	DGI	0.085s	0.134s	0.158s
	DGI_{BCE}	0.010s|8.5×	0.021s|6.4×	0.015s|10.5×

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

$$= \frac{1}{2N}\left(\sum_{i=1}^{N} \log(\boxed{\mathrm{sum}}(z_i)) + \log(1 - \boxed{\mathrm{sum}}(\tilde{z}_i))\right),$$

Replacing the summation with other aggregation function

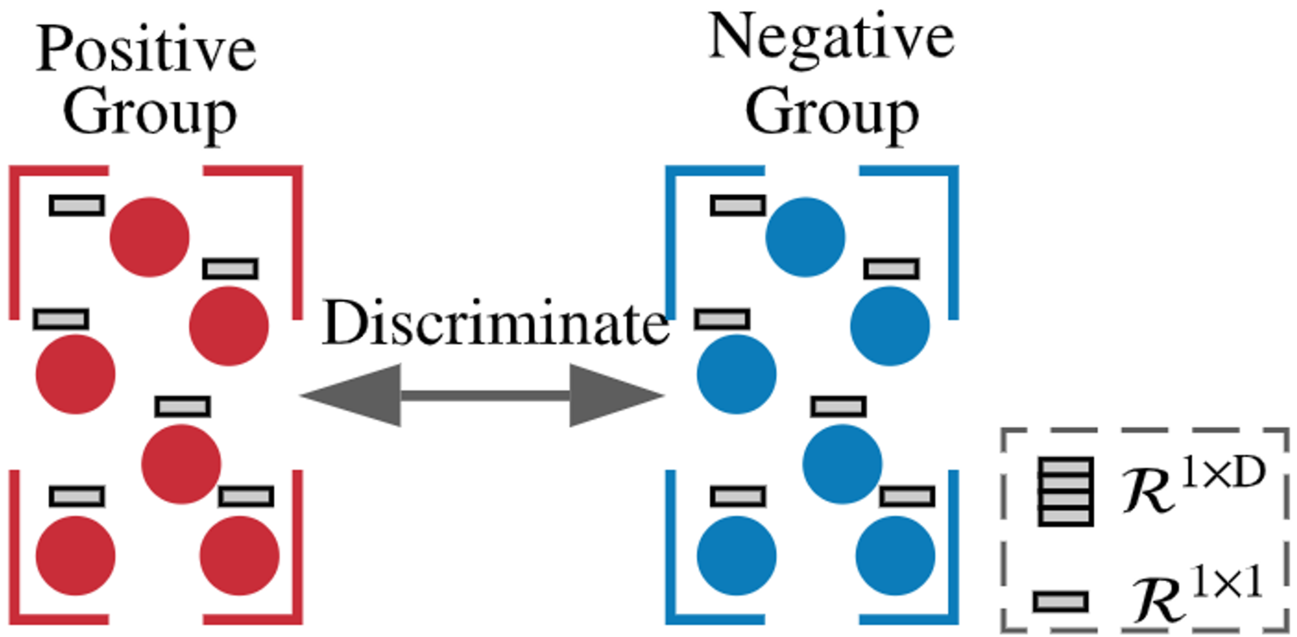
Table 11: The experiment result on three datasets with different aggregation function on node embeddings.

Method	Cora	CiteSeer	PubMed
Sum	82.5 ±0.2	71.7 ±0.6	77.7 ±0.5
Mean	81.8 ±0.5	71.8 ±1.1	76.5 ±1.2
Min	80.4 ±1.3	61.7 ±1.8	70.1 ±1.9
Max	71.4 ±1.2	65.3 ±1.4	70.2 ±2.8
linear	82.2 ±0.4	72.1 ±0.7	77.9 ±0.5

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Rethinking DGI

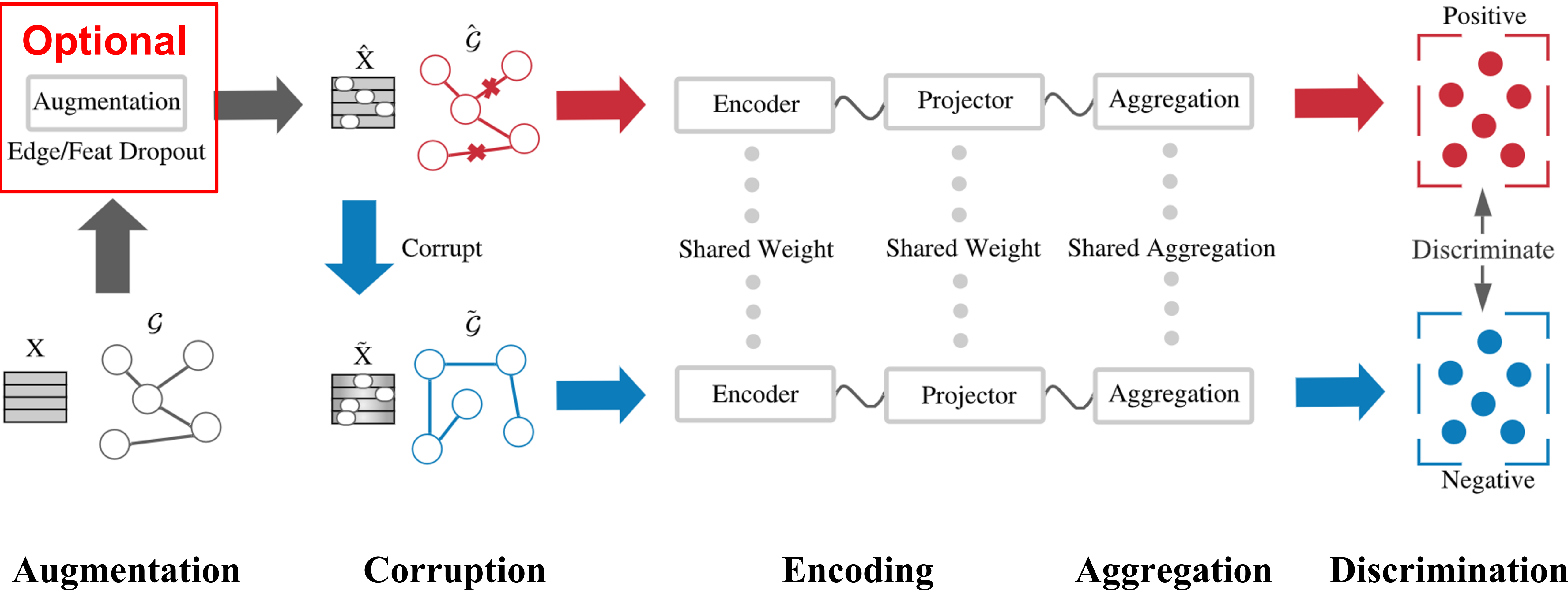
$$\mathcal{L}_{\mathrm{BCE}} = \frac{1}{2\mathrm{N}}\left(\sum_{\mathrm{i}=1}^{2\mathrm{N}} \mathrm{y_i} \log \mathrm{h_i} + (1 - \mathrm{y_i}) \log(1 - \mathrm{h_i})\right),$$



(b) Group Discrimination

With this loss, we can see Instead of contrastive learning, DGI is a Group Discrimination method

Proposed Framework: Graph Group Discrimination (GGD)



Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Experiment (Small-to-Medium scale Dataset)

Overall Performance Comparison

Data	Method	Cora	CiteSeer	PubMed	Comp	Photo
X, A, Y	GCN	81.5	70.3	79.0	76.3±0.5	87.3±1.0
X, A, Y	GAT	83.0±0.7	72.5±0.7	79.0±0.3	79.3±1.1	86.2±1.5
X, A, Y	SGC	81.0±0.0	71.9±0.1	78.9±0.0	74.4±0.1	86.4±0.0
X, A, Y	CG3	83.4±0.7	**73.6**±0.8	80.2±0.8	79.9±0.6	89.4±0.5
X, A	DGI	81.7±0.6	71.5±0.7	77.3±0.6	75.9±0.6	83.1±0.5
X, A	GMI	82.7±0.2	73.0±0.3	80.1±0.2	76.8±0.1	85.1±0.1
X, A	MVGRL	82.9±0.7	72.6±0.7	79.4±0.3	79.0±0.6	87.3±0.3
X, A	GRACE	80.0±0.4	71.7±0.6	79.5±1.1	71.8±0.4	81.8±1.0
X, A	BGRL	80.5±1.0	71.0±1.2	79.5±0.6	89.2±0.9	91.2±0.8
X, A	GBT	81.0±0.5	70.8±0.2	79.0±0.1	88.5±1.0	91.1±0.7
X, A	**GGD**	**84.1**±0.4	73.0±0.6	**81.3**±0.8	**90.1**±0.9	**92.5**±0.6

Time Consumption Improvement (epoch per second)

Method	Cora	CiteSeer	PubMed	Comp	Photo
DGI	0.085	0.134	0.158	0.171	0.059
GMI	0.394	0.497	2.285	1.297	0.637
MVGRL	0.123	0.171	0.488	0.663	0.468
GRACE	0.056	0.092	0.893	0.546	0.203
BGRL	0.085	0.094	0.147	0.337	0.273
GBT	0.073	0.072	0.103	0.492	0.173
GGD	0.010	0.021	0.015	0.016	0.009
Improve	7.3-39.4×	3.4-23.7×	6.9-152.3×	10.7-15.3×	19.2-70.8×

Memory Consumption Improvement (MB)

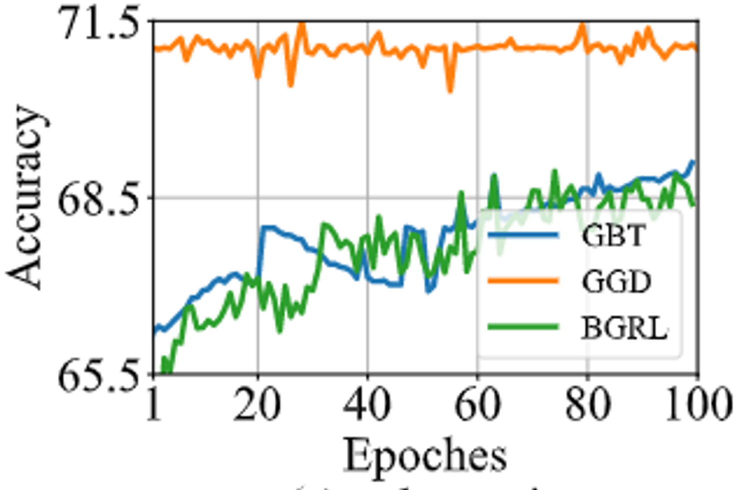
Method	Cora	CiteSeer	PubMed	Comp	Photo
DGI	4,189	8,199	11,471	7,991	4,946
GMI	4,527	5,467	14,697	10,655	5,219
MVGRL	5,381	5,429	6,619	6,645	6,645
GRACE	1,913	2,043	12,597	8,129	4,881
BGRL	1,627	1,749	2,299	5,069	3,303
GBT	1,651	1,799	2,461	5,037	2,641
GGD	1,475	1,587	1,629	1,787	1,637
Improve	10.7-72.6%	11.8-80.6%	27.2-85.8%	64.5-83.2%	38.0-75.4%

Experiment (Large scale Dataset - Ogbn-arxiv)

Using only 0.18 seconds and 69.8% less memory to reach SOTA.

10783 faster than existing methods.

Method	Valid	Test	Memory	Time	Total
Supervised GCN	73.0±0.2	71.7±0.3	-	-	-
MLP	57.7±0.4	55.5±0.2	-	-	-
Node2vec	71.3±0.1	70.1±0.1	-	-	-
DGI	71.3±0.1	70.3±0.2	-	-	-
GRACE(10k epos)	72.6±0.2	71.5±0.1	-	-	-
BGRL(10k epos)	72.5±0.1	71.6±0.1	OOM (Full-graph)	/	/
GBT(300 epos)	71.0±0.1	70.1±0.2	14,959MB	6.47	1,941.00
GGD(1 epo)	72.7±0.3	71.6±0.5	4,513MB|69.8%	0.18	0.18|10,783×



(a) ogbn-arxiv

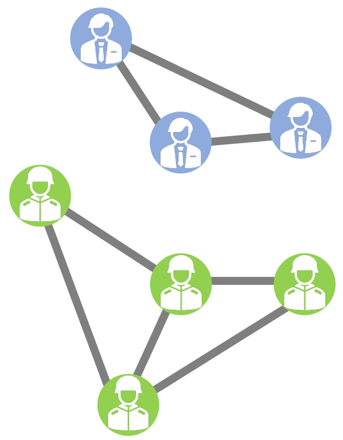
Fast convergence → converge with only 1 epoch

Zheng, Y., Pan, S., Lee, V., Zheng, Y., & Yu, P. S. (2022). Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, *35*, 10809-10820.

Heterophilic Graph Self-supervised Learning

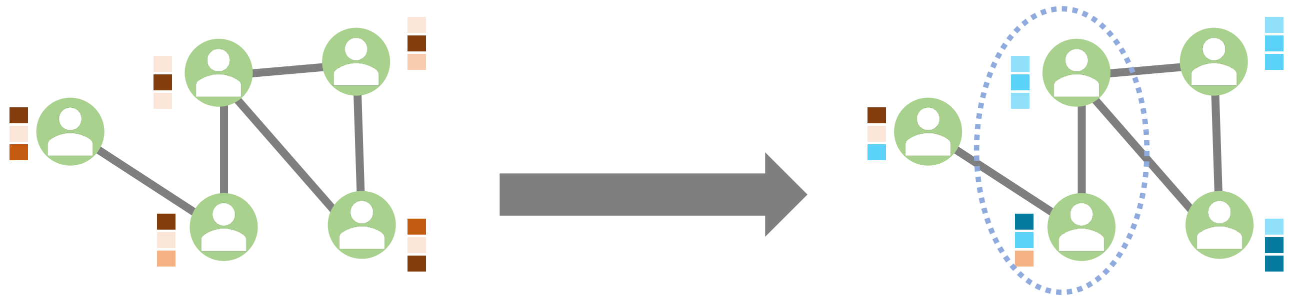
Homophily assumption

Most UGRL methods are designed based on the homophily assumption:



Linked nodes tend to share similar attributes with each other.

- Low-pass filter-like GNNs[1] (e.g., GCN) as encoders:



Representations of adjacent nodes become similar

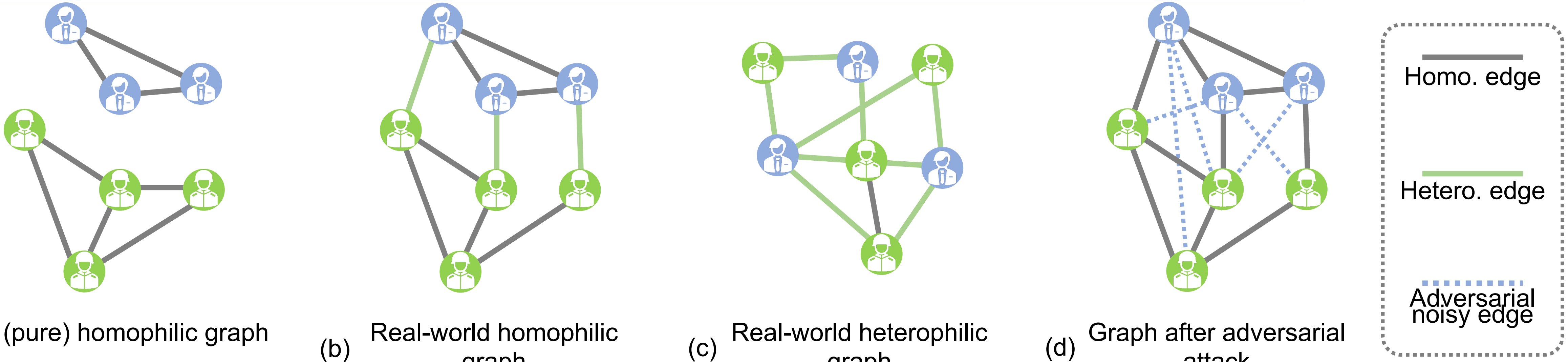
[1] Nt, H.; and Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. arXiv preprint arXiv:1905.09550..

Limitation

Do real-world graphs always obey the homophily assumption?
No!

- Pure homophilic graph is ideal, real-world graphs often contain heterophilic edges.
- Real-world homophilic graphs can also include heterophilic edges.
- In heterophilic graphs, heterophilic edges are much more than homophilic edges.
- Adversarial attack tends to reduce the homophily of graphs [5].

The behind homophily assumption hinders the generalization ability to heterophilic graphs and robustness against adversarial attack of most UGRL methods



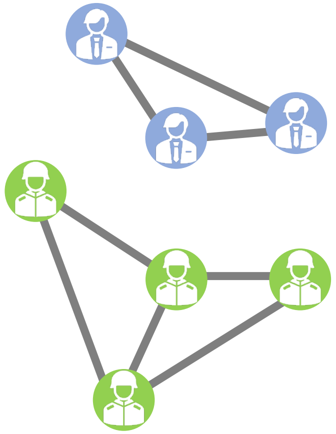
(a) Ideal (pure) homophilic graph (b) Real-world homophilic graph (c) Real-world heterophilic graph (d) Graph after adversarial attack

[5] Zhu J.; Jin J.; Loveland D.; Schaub, M.; and Koutra D. 2022. How does Heterophily Impact the Robustness of Graph Neural Networks?: Theoretical Connections and Practical Implications. In SIGKDD.

Observation

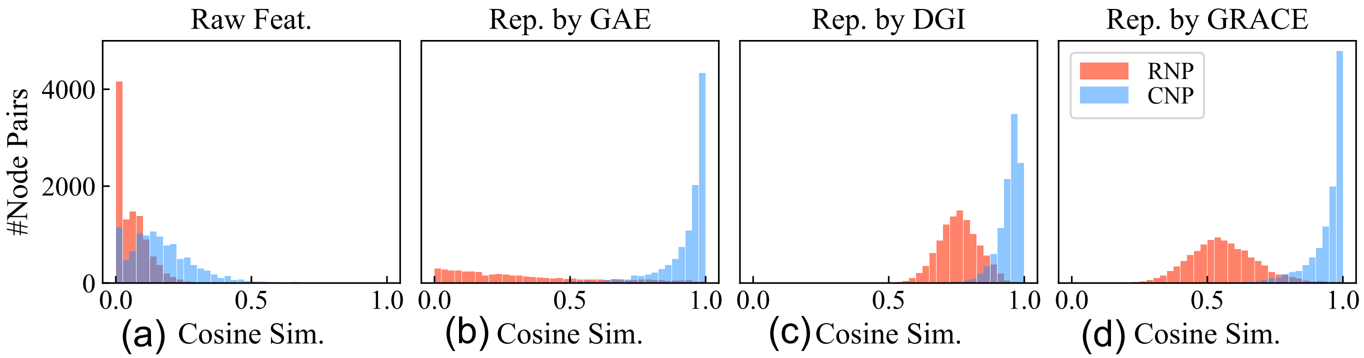
Most UGRL methods are designed based on the homophily assumption:

Linked nodes tend to share similar attributes with each other.

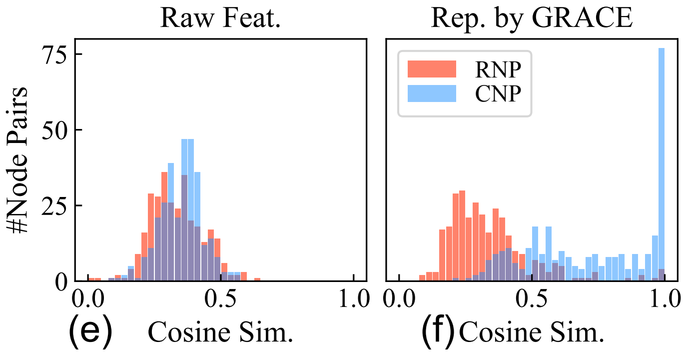


Visualization of the cosine similarity of:

CNP: connected node pairs
RNP: randomly sampled node pairs



Cora dataset



Texas dataset

All the connected nodes are pushed to be closer in the representation space, even if some of them have moderate feature similarities that are comparable to randomly sampled node pairs.

Contribution

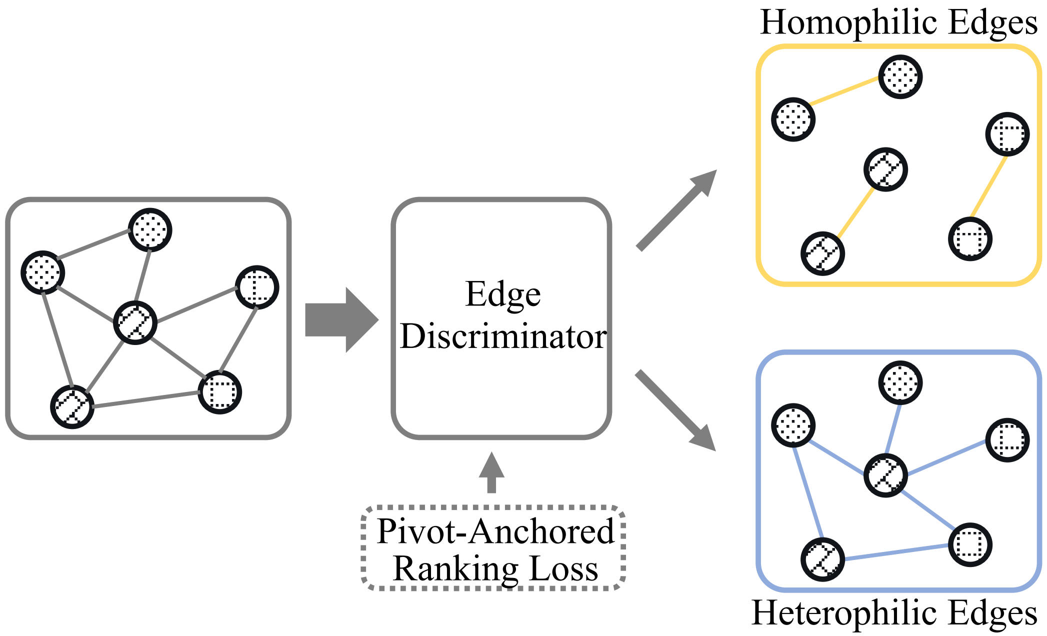
To address the aforementioned limitation…

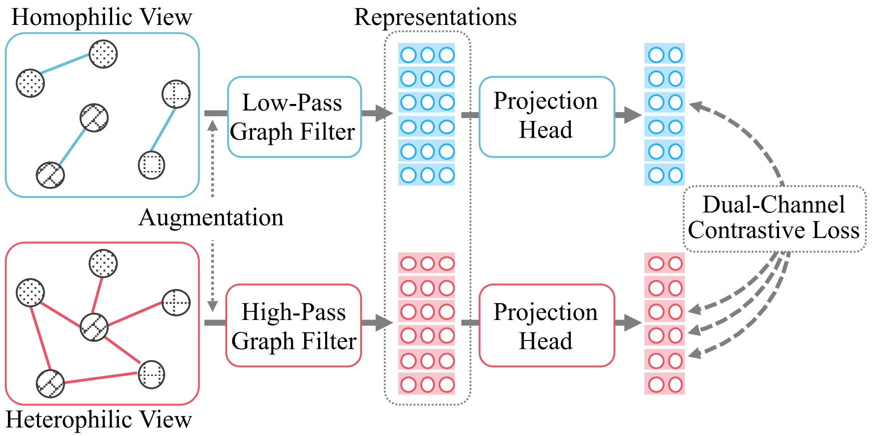
(Q1) Is it possible to distinguish between two types of edges in an unsupervised manner?

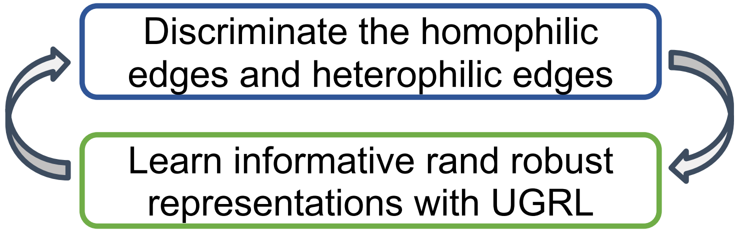
(A1) trainable edge discriminator with a pivot-anchored ranking loss function.

(Q2) How to effectively couple edge discriminating with representation learning into an integrated UGRL model?

(A2) dual-channel graph encoding module with robust cross-channel contrasting.
Training with a closed-loop interplay.

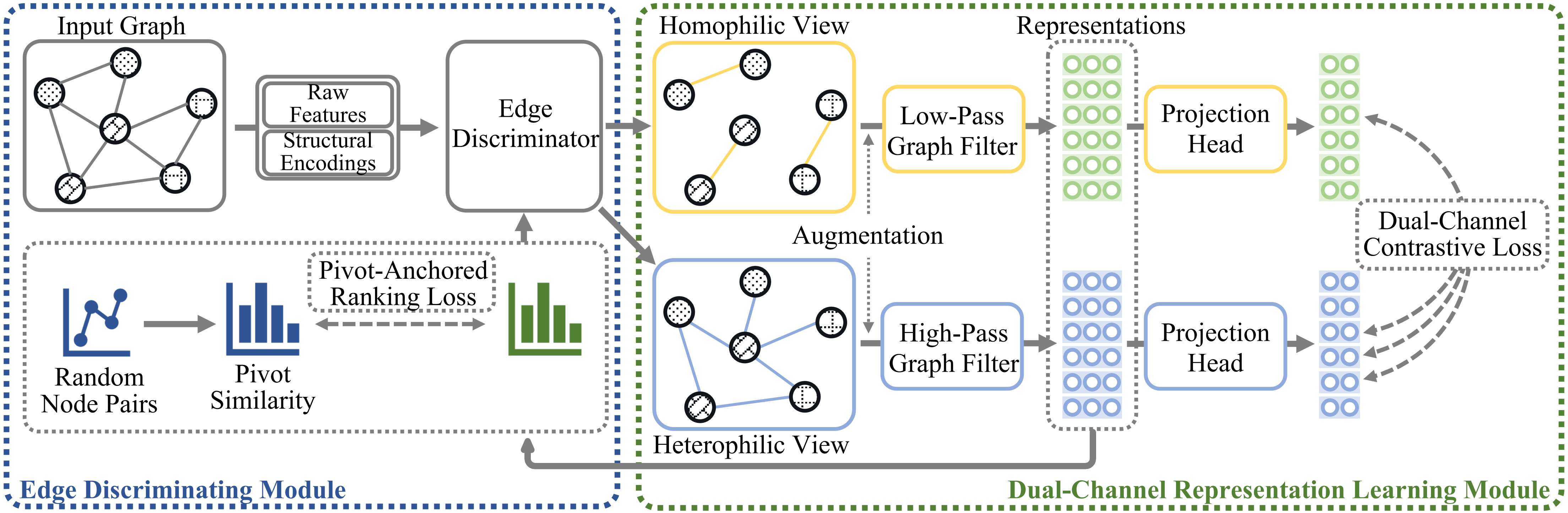






Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065*.

Proposed method - GREET

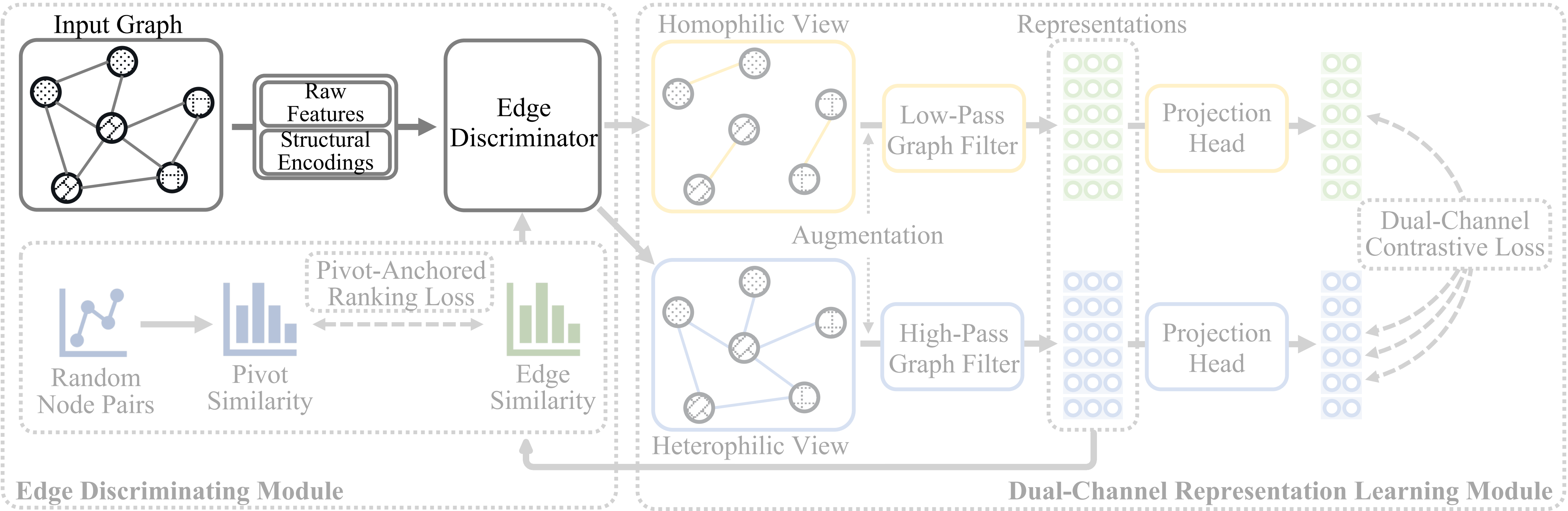


To discriminates the homophilic and heterophilic edges without accessing node labels.

To leverage both types of edges to generate informative node representations.

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065.*

Edge discriminating



- Edge discriminator – a two-layer MLP:

$$\mathbf{h}'_i = \mathrm{MLP}_1([\mathbf{x}_i \| \mathbf{s}_i]),\ \mathbf{h}'_j = \mathrm{MLP}_1([\mathbf{x}_j \| \mathbf{s}_j]),$$
$$\theta_{i,j} = \left(\mathrm{MLP}_2([\mathbf{h}'_i \| \mathbf{h}'_j]) + \mathrm{MLP}_2([\mathbf{h}'_j \| \mathbf{h}'_i])\right)/2,$$

- Input of edge discriminator:

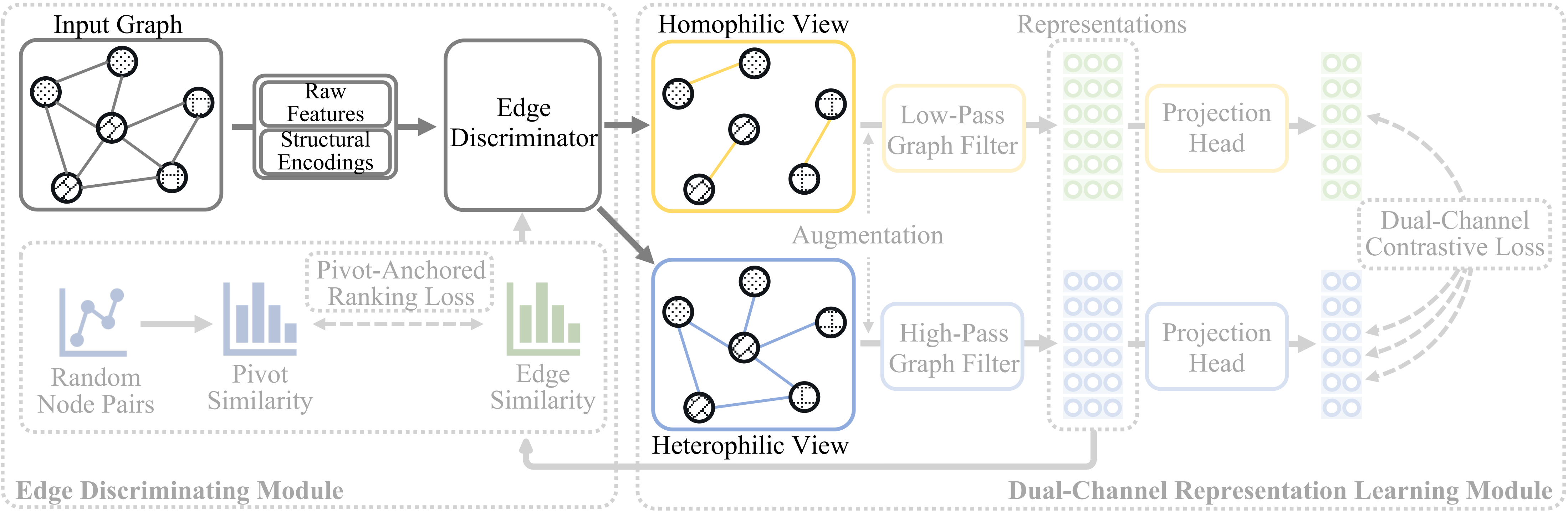
Raw feature + Structural encoding (SE)

Random walk diffusion process-based SE [6]:

$$\mathbf{s}_i = \left[\mathbf{T}_{ii}, \mathbf{T}^2_{ii}, \cdots, \mathbf{T}^{d_s}_{ii}\right] \in \mathbb{R}^{d_s} \text{ where } \mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$$

[6] Dwivedi, V. P.; Luu, A. T.; Laurent, T.; Bengio, Y.; and Bresson, X. 2022. Graph Neural Networks with Learnable Struc- tural and Positional Representations. In ICLR.

View generalization



- Gumbel-Max reparametrization trick [7]:

$$\hat{w}_{i,j} = \mathrm{Sigmoid}\Big(\big(\theta_{i,j} + \log\delta - \log(1-\delta)\big)/\tau_g\Big)$$

- View generation:

Input graph: $\mathcal{G} = (\mathbf{A}, \mathbf{X})$

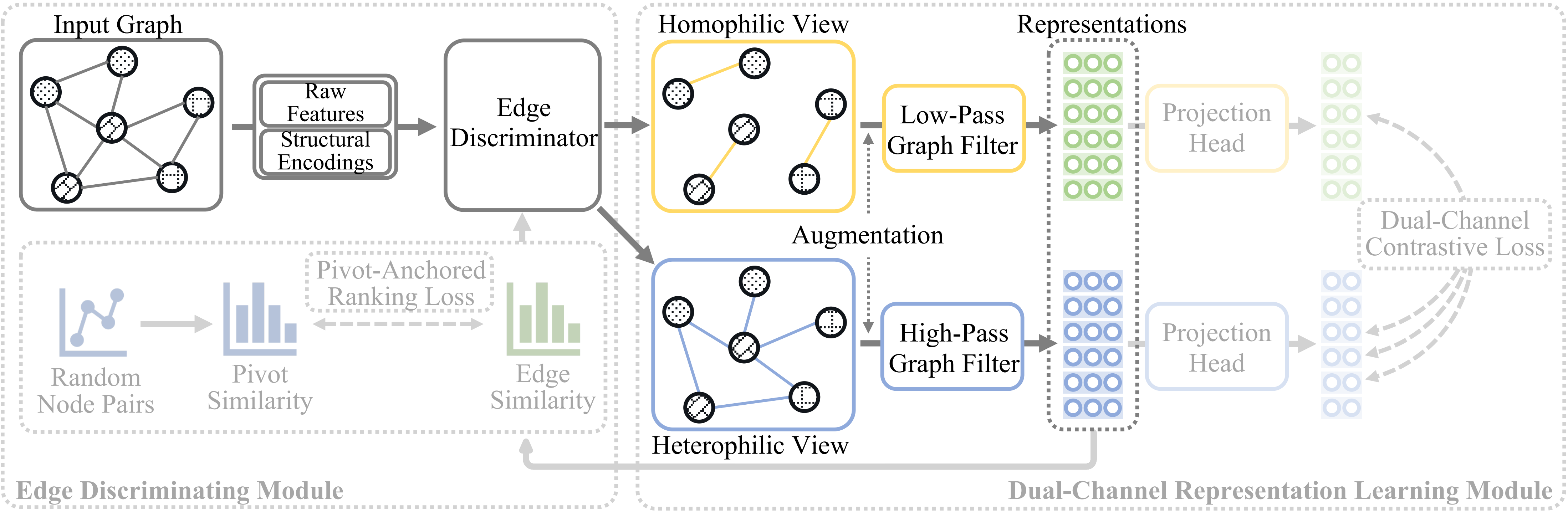
Homo. view $\mathcal{G}^{(hm)} = (\mathbf{A}^{(hm)}, \mathbf{X})$

Hetero. view $\mathcal{G}^{(ht)} = (\mathbf{A}^{(ht)}, \mathbf{X})$

where $\mathbf{A}^{(hm)}_{i,j} = \hat{w}_{i,j},\ \mathbf{A}^{(ht)}_{i,j} = 1 - \hat{w}_{i,j},\ \text{for } e_{i,j} \in \mathcal{E}$

[7] Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In ICLR.

Dual-channel encoding



- Homo. View encoder – low-pass filter:

$$\mathbf{H}_0^{(hm)} = \mathrm{MLP}^{(hm)}(\mathbf{X}),\ \mathbf{H}_l^{(hm)} = \tilde{\mathbf{A}}^{(hm)}\mathbf{H}_{l-1}^{(hm)}$$

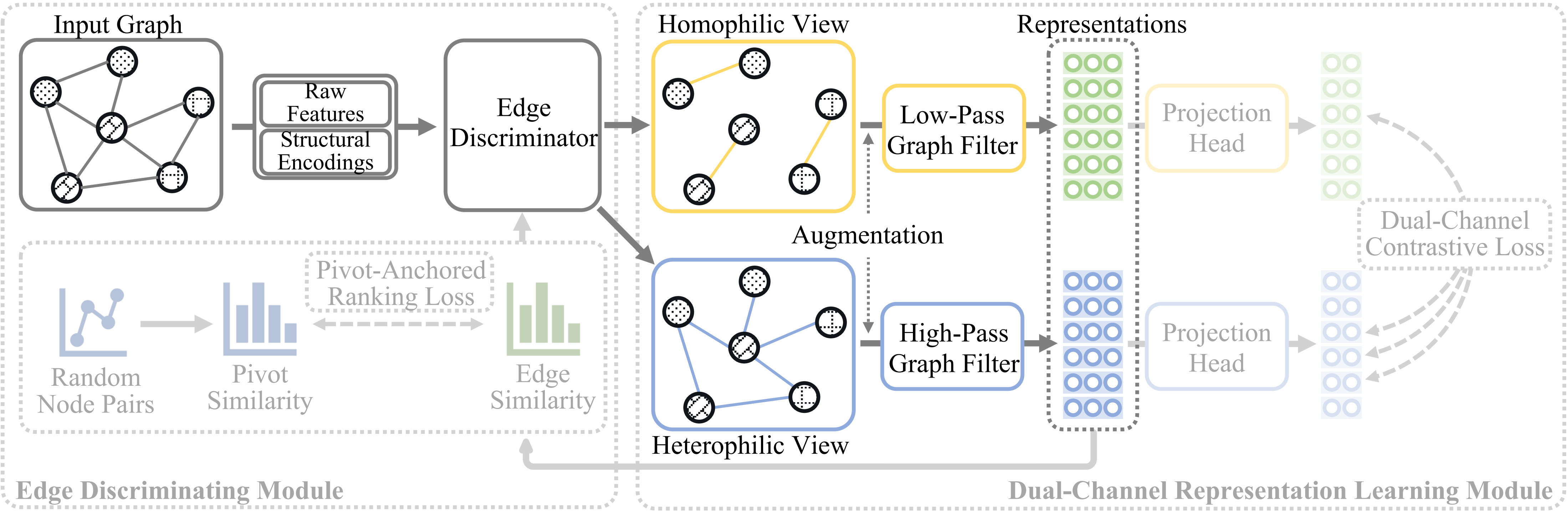
- Hetero. View encoder – high-pass filter:

$$\mathbf{H}_0^{(ht)} = \mathrm{MLP}^{(ht)}(\mathbf{X}),\ \mathbf{H}_l^{(ht)} = \tilde{\mathbf{L}}^{(ht)}\mathbf{H}_{l-1}^{(ht)},$$

where $\tilde{\mathbf{L}}^{(ht)} = \mathbf{I} - \alpha\tilde{\mathbf{A}}^{(ht)}$

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065*.

Dual-channel encoding



- Homo. View encoder – low-pass filter:

$$\mathbf{H}_0^{(hm)} = \mathrm{MLP}^{(hm)}(\mathbf{X}),\ \mathbf{H}_l^{(hm)} = \tilde{\mathbf{A}}^{(hm)}\mathbf{H}_{l-1}^{(hm)}$$

- Hetero. View encoder – low-pass filter:

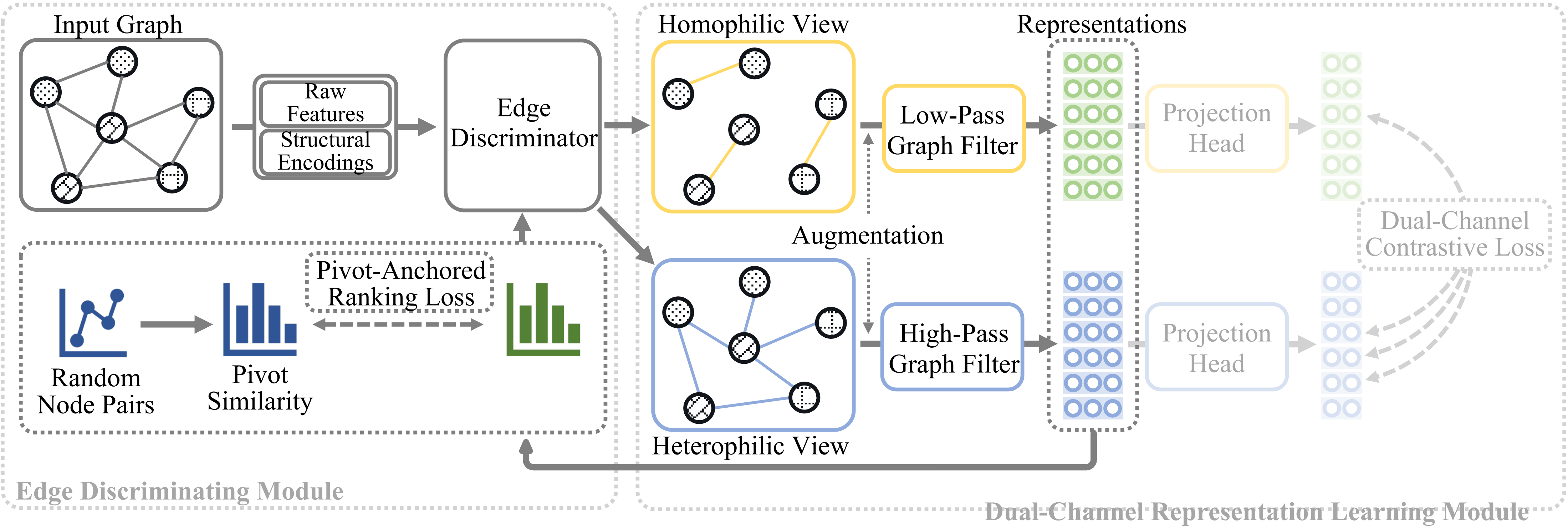
$$\mathbf{H}_0^{(ht)} = \mathrm{MLP}^{(ht)}(\mathbf{X}),\ \mathbf{H}_l^{(ht)} = \tilde{\mathbf{L}}^{(ht)}\mathbf{H}_{l-1}^{(ht)},$$

concat

Node representations:

$$\mathbf{H} = [\mathbf{H}^{(hm)} \| \mathbf{H}^{(ht)}] \in \mathbb{R}^{n \times d_r}$$

Pivot-anchored ranking loss



$$\mathcal{R}^{(hm)}(e_{i,j}) = [\boxed{s_{v_{i'},v_{j'}}} - \boxed{s_{e_{i,j}}} + \gamma^{(hm)}]_+,$$

$$\mathcal{R}^{(ht)}(e_{i,j}) = [s_{e_{i,j}} - s_{v_{i'},v_{j'}} + \gamma^{(ht)}]_+,$$

Where $s_{e_{i,j}} = \cos(\mathbf{h}_i, \mathbf{h}_j)$

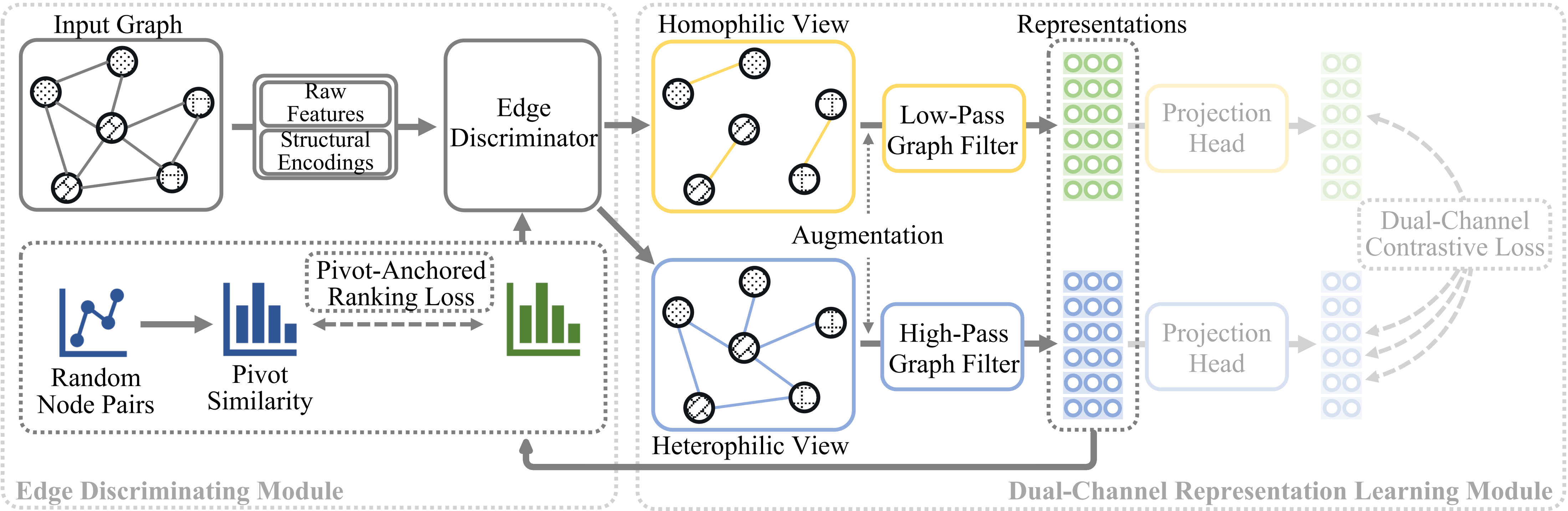
Rep. sim. of connected nodes i, j where $e_{i,j}$ is an existing edge

Rep. sim. of two randomly sampled nodes v_i', v_j'

$\gamma^{(hm)}, \gamma^{(ht)}$: margins (hyper-params)

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065.*

Pivot-anchored ranking loss



$$\mathcal{R}^{(hm)}(e_{i,j}) = [s_{v_{i'},v_{j'}} - s_{e_{i,j}} + \gamma^{(hm)}]_+,$$

$$\mathcal{R}^{(ht)}(e_{i,j}) = [s_{e_{i,j}} - s_{v_{i'},v_{j'}} + \gamma^{(ht)}]_+,$$

Where $s_{e_{i,j}} = \cos(\mathbf{h}_i, \mathbf{h}_j)$

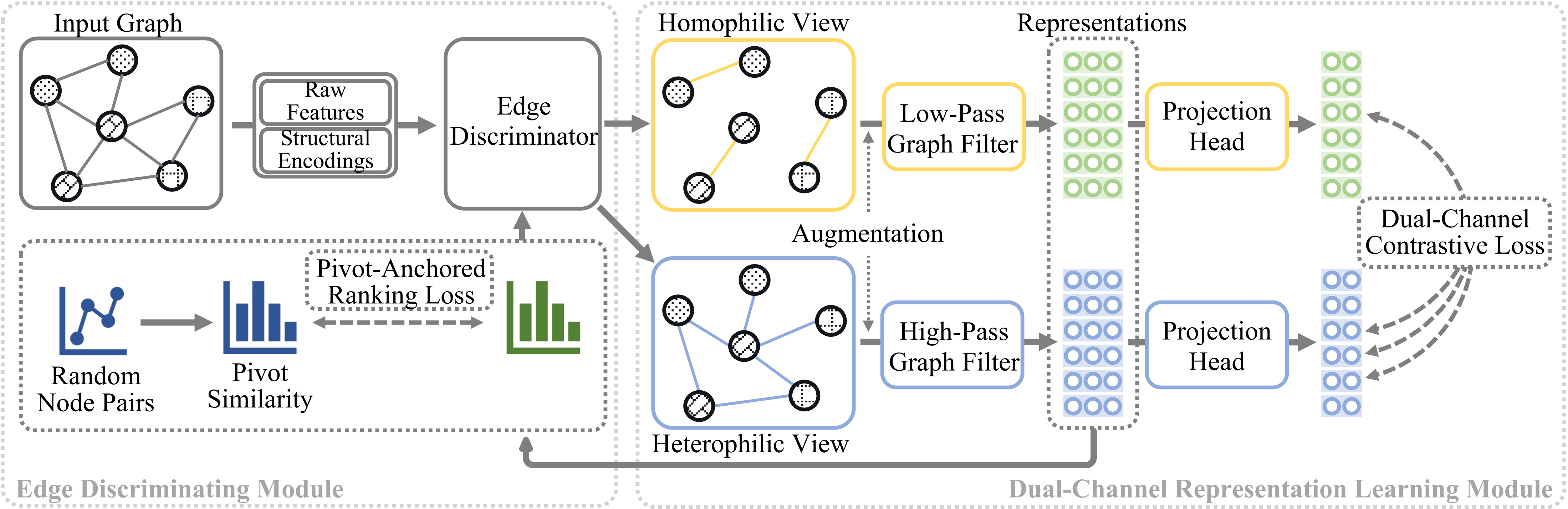
$$\mathcal{L}_r = \mathcal{L}_r^{(hm)} + \mathcal{L}_r^{(ht)}$$

$$\mathcal{L}_r^{(hm)} = \mathop{\mathbb{E}}_{e_{i,j} \sim \Pi(\hat{w}_{i,j})} \mathcal{R}^{(hm)}(e_{i,j}) = \frac{1}{\hat{W}^{(hm)}} \sum_{e_{i,j} \in \mathcal{E}} \hat{w}_{i,j} \mathcal{R}^{(hm)}(e_{i,j}),$$

$$\mathcal{L}_r^{(ht)} = \mathop{\mathbb{E}}_{e_{i,j} \sim \Pi(1-\hat{w}_{i,j})} \mathcal{R}^{(ht)}(e_{i,j}) = \frac{1}{\hat{W}^{(ht)}} \sum_{e_{i,j} \in \mathcal{E}} (1-\hat{w}_{i,j}) \mathcal{R}^{(ht)}(e_{i,j}),$$

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065.*

Dual-channel contrastive loss



kNN extends positive samples:

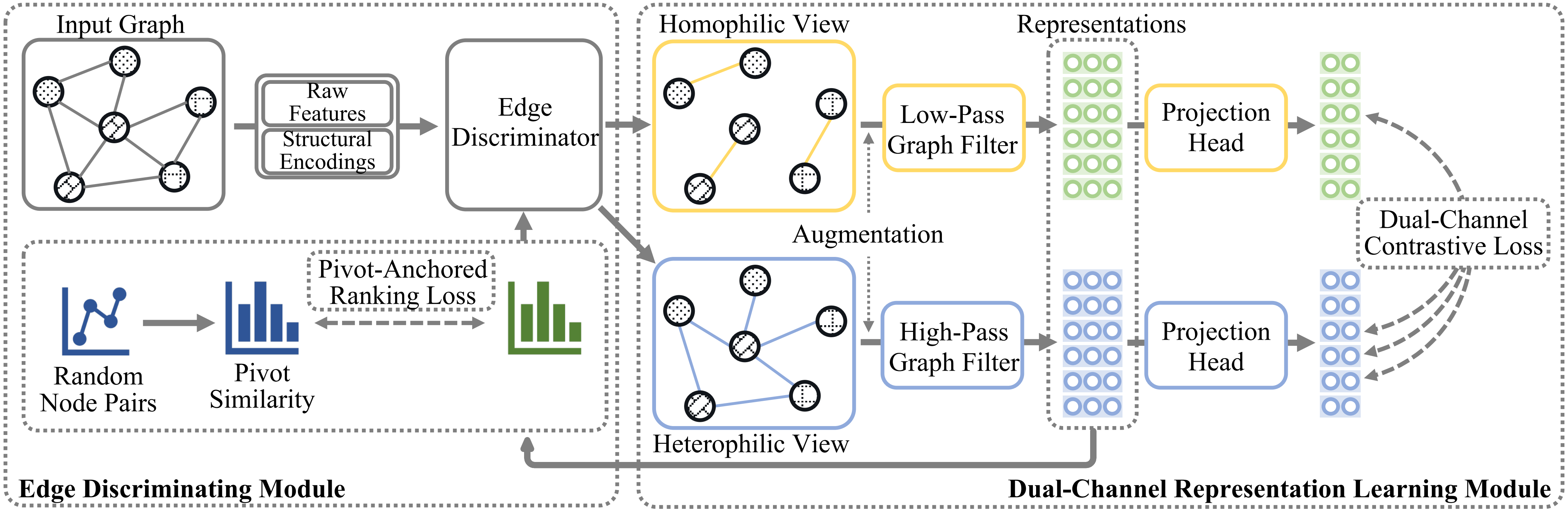
$$\mathcal{N}_i = \mathrm{kNN}(v_i, k)$$

$$\mathcal{L}_c = -\frac{1}{n}\sum_{v_i \in \mathcal{V}}\left[\frac{1}{2|\mathcal{N}_i|}\sum_{v_j \in \mathcal{N}_i}\left(\log \frac{e^{\cos\left(\mathbf{z}_i^{(hm)}, \mathbf{z}_j^{(ht)}\right)/\tau_c}}{\sum_{v_k \in \mathcal{V}\setminus v_i} e^{\cos\left(\mathbf{z}_i^{(hm)}, \mathbf{z}_k^{(ht)}\right)/\tau_c}} + \log \frac{e^{\cos\left(\mathbf{z}_i^{(ht)}, \mathbf{z}_j^{(hm)}\right)/\tau_c}}{\sum_{v_k \in \mathcal{V}\setminus v_i} e^{\cos\left(\mathbf{z}_i^{(ht)}, \mathbf{z}_k^{(hm)}\right)/\tau_c}}\right)\right],$$

(Cross-view Info-NCE loss)

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065.*

Alternative training scheme



Train Representation Learning Module

Better edge discrimination

Better node representations

Train Edge Discriminating Module

Overall optimization objective:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_c$$

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065.*

Performance comparison

- Node classification @ homophilic graphs

Methods	Cora	CiteSeer	PubMed	Wiki-CS	Amz. Comp.	Amz. Photo	Co. CS	Co. Physics
GCN*	81.5	70.3	79.0	76.89±0.37	86.34±0.48	92.35±0.25	93.10±0.17	95.54±0.19
GAT*	83.0	72.5	79.0	77.42±0.19	87.06±0.35	92.64±0.42	92.41±0.27	95.45±0.17
MLP	56.11±0.34	56.91±0.42	71.35±0.05	72.02±0.21	73.88±0.10	78.54±0.05	90.42±0.08	93.54±0.05
DeepWalk	69.47±0.55	58.82±0.61	69.87±1.25	74.35±0.06	85.68±0.06	89.44±0.11	84.61±0.22	91.77±0.15
node2vec	71.24±0.89	47.64±0.77	66.47±1.00	71.79±0.05	84.39±0.08	89.67±0.12	85.08±0.03	91.19±0.04
GAE	71.07±0.39	65.22±0.43	71.73±0.92	70.15±0.01	85.27±0.19	91.62±0.13	90.01±0.71	94.92±0.07
VGAE	79.81±0.87	66.75±0.37	77.16±0.31	75.63±0.19	86.37±0.21	92.20±0.11	92.11±0.09	94.52±0.00
DGI	82.29±0.56	71.49±0.14	77.43±0.84	75.73±0.13	84.09±0.39	91.49±0.25	91.95±0.40	94.57±0.38
GMI	82.51±1.47	71.56±0.56	79.83±0.90	75.06±0.13	81.76±0.52	90.72±0.33	OOM	OOM
MVGRL	83.03±0.27	72.75±0.46	79.63±0.38	77.97±0.18	87.09±0.27	92.01±0.13	91.97±0.19	95.53±0.10
GRACE	80.08±0.53	71.41±0.38	80.15±0.34	79.16±0.36	87.21±0.44	92.65±0.32	92.78±0.23	95.39±0.32
GCA	80.39±0.42	71.21±0.24	**80.37±0.75**	79.35±0.12	87.84±0.27	92.78±0.17	93.32±0.12	95.87±0.15
BGRL	81.08±0.17	71.59±0.42	79.97±0.36	78.74±0.22	**88.92±0.33**	**93.24±0.29**	93.26±0.36	95.76±0.38
GREET	**83.81±0.87**	**73.08±0.84**	80.29±1.00	**80.68±0.31**	87.94±0.35	92.85±0.31	**94.65±0.18**	**96.13±0.12**

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065*.

Performance comparison

- Node classification @ heterophilic graphs

Methods	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin
GCN	59.63±2.32	36.28±1.52	30.83±0.77	57.03±3.30	60.00±4.80	56.47±6.55
GAT	56.38±2.19	32.09±3.27	28.06±1.48	59.46±3.63	61.62±3.78	54.71±6.87
MLP	46.91±2.15	29.28±1.33	35.66±0.94	81.08±7.93	81.62±5.51	84.31±3.40
Geom-GCN*	60.90	38.14	31.63	60.81	67.57	64.12
H2GCN*	59.39±1.98	37.90±2.02	<u>35.86±1.03</u>	<u>82.16±4.80</u>	<u>84.86±6.77</u>	**86.67±4.69**
FAGCN	<u>63.44±2.05</u>	<u>41.17±1.94</u>	35.74±0.62	81.35±5.05	84.32±6.02	83.33±2.01
GPR-GNN	61.58±2.24	39.65±2.81	35.27±1.04	81.89±5.93	83.24±4.95	84.12±3.45
DeepWalk	47.74±2.05	32.93±1.58	22.78±0.64	39.18±5.57	46.49±6.49	33.53±4.92
node2vec	41.93±3.29	22.84±0.72	28.28±1.27	42.94±7.46	41.92±7.76	37.45±7.09
GAE	33.84±2.77	28.03±1.61	28.03±1.18	58.85±3.21	58.64±4.53	52.55±3.80
VGAE	35.22±2.71	29.48±1.48	26.99±1.56	59.19±4.09	59.20±4.26	56.67±5.51
DGI	39.95±1.75	31.80±0.77	29.82±0.69	63.35±4.61	60.59±7.56	55.41±5.96
GMI	46.97±3.43	30.11±1.92	27.82±0.90	54.76±5.06	50.49±2.21	45.98±2.76
MVGRL	51.07±2.68	35.47±1.29	30.02±0.70	64.30±5.43	62.38±5.61	62.37±4.32
GRACE	48.05±1.81	31.33±1.22	29.01±0.78	54.86±6.95	57.57±5.68	50.00±5.83
GRACE-FA	52.68±2.14	35.97±1.20	32.55±1.28	67.57±4.98	64.05±7.46	63.73±6.81
GCA	49.80±1.81	35.50±0.91	29.65±1.47	55.41±4.56	59.46±6.16	50.78±4.06
BGRL	47.46±2.74	32.64±0.78	29.86±0.75	57.30±5.51	59.19±5.85	52.35±4.12
GREET	**63.64±1.26**	**42.29±1.43**	**36.55±1.01**	**85.14±4.87**	**87.03±2.36**	<u>84.90±4.48</u>

Liu, Y., Zheng, Y., Zhang, D., Lee, V., & Pan, S. (2022). Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. *arXiv preprint arXiv:2211.14065*.

Heterogenous Graph Self-supervised Learning

Heterogeneous Graphs

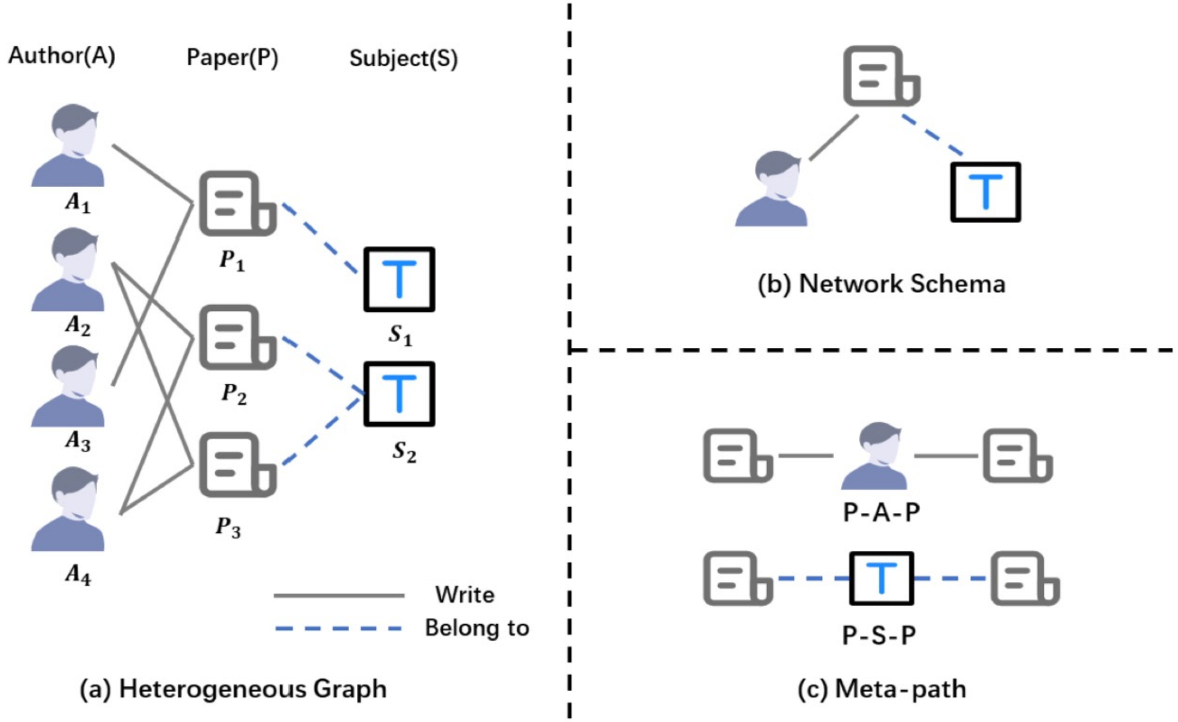


Figure 1: A toy example of HIN (ACM) and relative illustrations of meta-path and network schema.

Heterogeneous Graph has
different types of nodes or edges

Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

HeCo Framework – (View Generation)

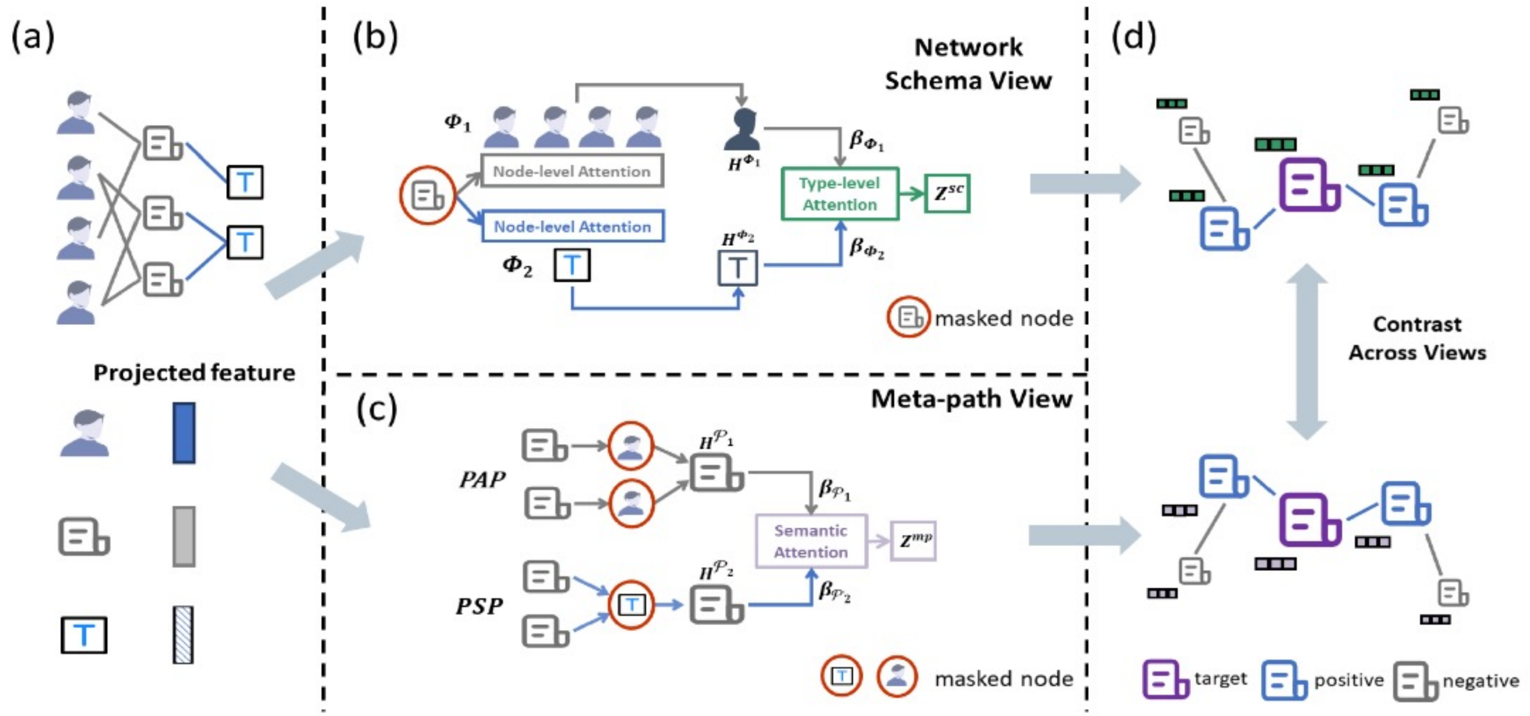


Figure 2: The overall architecture of our proposed HeCo.

Network Schema View

$$h_i^{\Phi_m} = \sigma\left(\sum_{j\in N_i^{\Phi_m}} \alpha_{i,j}^{\Phi_m} \cdot h_j\right)$$

$$\alpha_{i,j}^{\Phi_m} = \frac{\exp\left(LeakyReLU\left(\mathbf{a}_{\Phi_m}^{\top} \cdot [h_i||h_j]\right)\right)}{\sum_{l\in N_i^{\Phi_m}} \exp\left(LeakyReLU\left(\mathbf{a}_{\Phi_m}^{\top} \cdot [h_i||h_l]\right)\right)},$$

Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

HeCo Framework – (View Generation)

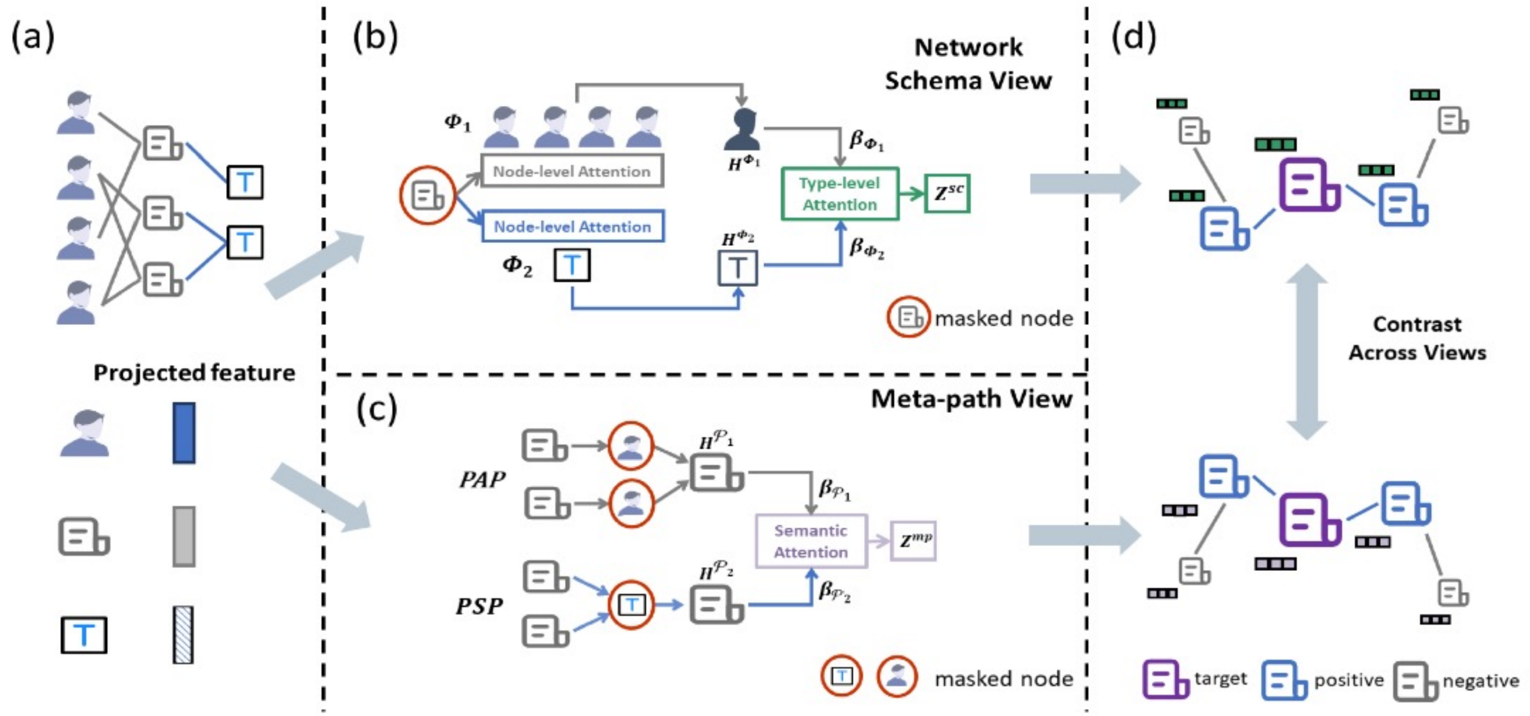


Figure 2: The overall architecture of our proposed HeCo.

Meta-path View

$$w_{\Phi_m} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{sc}^{\top} \cdot \tanh\left(\mathbf{W}_{sc} h_i^{\Phi_m} + \mathbf{b}_{sc}\right),$$

$$\beta_{\Phi_m} = \frac{\exp\left(w_{\Phi_m}\right)}{\sum_{i=1}^{S} \exp\left(w_{\Phi_i}\right)},$$

$$z_i^{sc} = \sum_{m=1}^{S} \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$

Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

HeCo Framework – (Contrastive Learning)

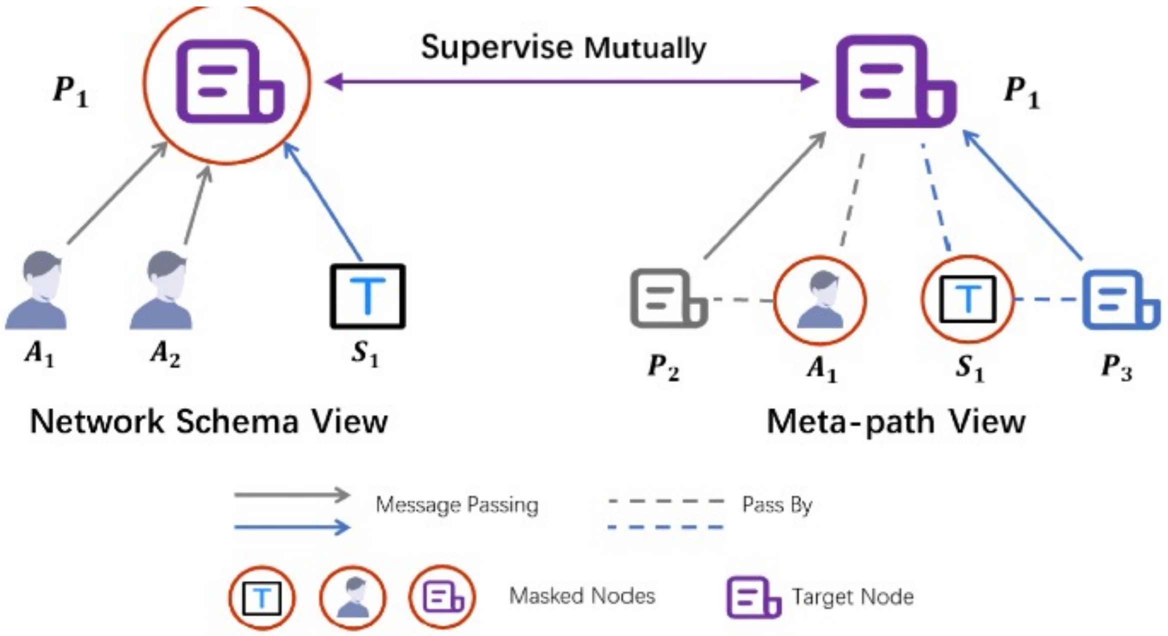


Figure 3: A schematic diagram of view mask mechanism.

Masked Node in Network Schema View

Masked People/Subjects in Meta-path View

$$z_i^{sc}\_proj = W^{(2)}\sigma\left(W^{(1)}z_i^{sc} + b^{(1)}\right) + b^{(2)},$$
$$z_i^{mp}\_proj = W^{(2)}\sigma\left(W^{(1)}z_i^{mp} + b^{(1)}\right) + b^{(2)},$$

$$\mathcal{L}_i^{sc} = -\log \frac{\sum_{j\in\mathbb{P}_i} exp\left(sim\left(z_i^{sc}\_proj, z_j^{mp}\_proj\right)/\tau\right)}{\sum_{k\in\{\mathbb{P}_i \cup \mathbb{N}_i\}} exp\left(sim\left(z_i^{sc}\_proj, z_k^{mp}\_proj\right)/\tau\right)},$$

$$\mathcal{J} = \frac{1}{|V|}\sum_{i\in V}\left[\lambda \cdot \mathcal{L}_i^{sc} + (1-\lambda)\cdot \mathcal{L}_i^{mp}\right],$$

Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

HeCo Framework – (Contrastive Learning)

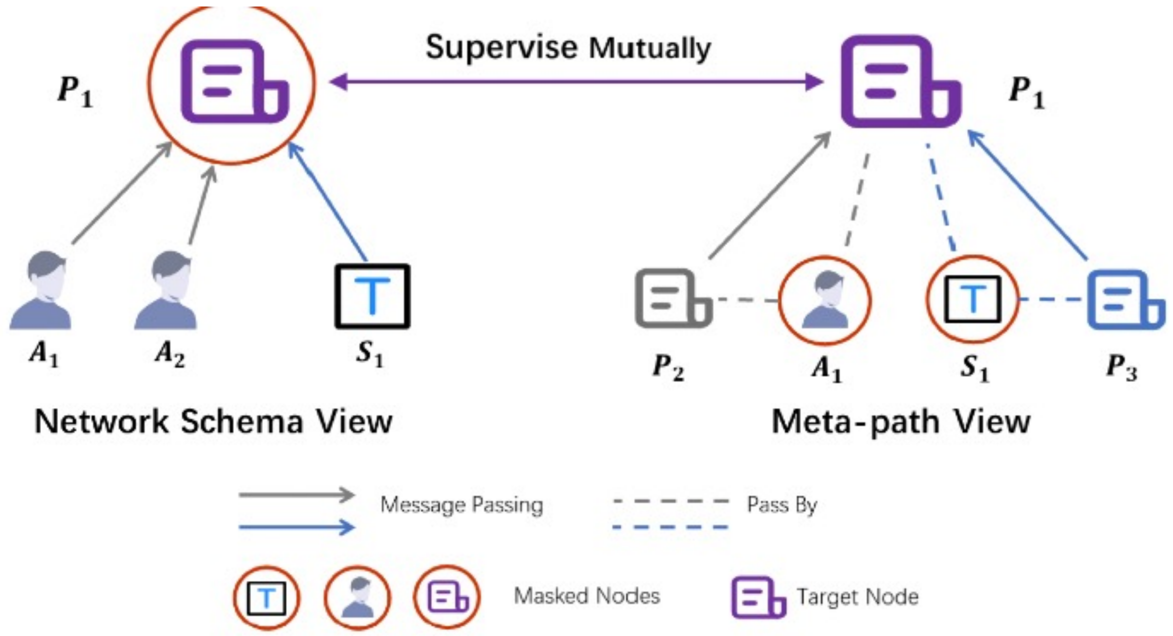


Figure 3: A schematic diagram of view mask mechanism.

$$z_i^{sc}\_proj = W^{(2)}\sigma\left(W^{(1)}z_i^{sc} + b^{(1)}\right) + b^{(2)},$$
$$z_i^{mp}\_proj = W^{(2)}\sigma\left(W^{(1)}z_i^{mp} + b^{(1)}\right) + b^{(2)},$$

$$\mathcal{L}_i^{sc} = -\log \frac{\sum_{j\in\mathbb{P}_i} exp\left(sim\left(z_i^{sc}\_proj, z_j^{mp}\_proj\right)/\tau\right)}{\sum_{k\in\{\mathbb{P}_i \bigcup \mathbb{N}_i\}} exp\left(sim\left(z_i^{sc}\_proj, z_k^{mp}\_proj\right)/\tau\right)},$$

We can obtain L^{mp} similarly

$$\mathcal{J} = \frac{1}{|V|}\sum_{i\in V}\left[\lambda \cdot \mathcal{L}_i^{sc} + (1-\lambda)\cdot \mathcal{L}_i^{mp}\right],$$

Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

Experiment

Table 2: Quantitative results (%±σ) on node classification.

Datasets	Metric	Split	GraphSAGE	GAE	Mp2vec	HERec	HetGNN	HAN	DGI	DMGI	HeCo
ACM	Ma-F1	20	47.13±4.7	62.72±3.1	51.91±0.9	55.13±1.5	72.11±0.9	85.66±2.1	79.27±3.8	87.86±0.2	**88.56±0.8**
		40	55.96±6.8	61.61±3.2	62.41±0.6	61.21±0.8	72.02±0.4	87.47±1.1	80.23±3.3	86.23±0.8	**87.61±0.5**
		60	56.59±5.7	61.67±2.9	61.13±0.4	64.35±0.8	74.33±0.6	88.41±1.1	80.03±3.3	87.97±0.4	**89.04±0.5**
	Mi-F1	20	49.72±5.5	68.02±1.9	53.13±0.9	57.47±1.5	71.89±1.1	85.11±2.2	79.63±3.5	87.60±0.8	**88.13±0.8**
		40	60.98±3.5	66.38±1.9	64.43±0.6	62.62±0.9	74.46±0.8	87.21±1.2	80.41±3.0	86.02±0.9	**87.45±0.5**
		60	60.72±4.3	65.71±2.2	62.72±0.3	65.15±0.9	76.08±0.7	88.10±1.2	80.15±3.2	87.82±0.5	**88.71±0.5**
	AUC	20	65.88±3.7	79.50±2.4	71.66±0.7	75.44±1.3	84.36±1.0	93.47±1.5	91.47±2.3	**96.72±0.3**	96.49±0.3
		40	71.06±5.2	79.14±2.5	80.48±0.4	79.84±0.5	85.01±0.6	94.84±0.9	91.52±2.3	96.35±0.3	**96.40±0.4**
		60	70.45±6.2	77.90±2.8	79.33±0.4	81.64±0.7	87.64±0.7	94.68±1.4	91.41±1.9	**96.79±0.2**	96.55±0.3
DBLP	Ma-F1	20	71.97±8.4	90.90±0.1	88.98±0.2	89.57±0.4	89.51±1.1	89.31±0.9	87.93±2.4	89.94±0.4	**91.28±0.2**
		40	73.69±8.4	89.60±0.3	88.68±0.2	89.73±0.4	88.61±0.8	88.87±1.0	88.62±0.6	89.25±0.4	**90.34±0.3**
		60	73.86±8.1	90.08±0.2	90.25±0.1	90.18±0.3	89.56±0.5	89.20±0.8	89.19±0.9	89.46±0.6	**90.64±0.3**
	Mi-F1	20	71.44±8.7	91.55±0.1	89.67±0.1	90.24±0.4	90.11±1.0	90.16±0.9	88.72±2.6	90.78±0.3	**91.97±0.2**
		40	73.61±8.6	90.00±0.3	89.14±0.2	90.15±0.4	89.03±0.7	89.47±0.9	89.22±0.5	89.92±0.4	**90.76±0.3**
		60	74.05±8.3	90.95±0.2	91.17±0.1	91.01±0.3	90.43±0.6	90.34±0.8	90.35±0.8	90.66±0.5	**91.59±0.2**
	AUC	20	90.59±4.3	98.15±0.1	97.69±0.0	98.21±0.2	97.96±0.4	98.07±0.6	96.99±1.4	97.75±0.3	**98.32±0.1**
		40	91.42±4.0	97.85±0.1	97.08±0.0	97.93±0.1	97.70±0.3	97.48±0.6	97.12±0.4	97.23±0.2	**98.06±0.1**
		60	91.73±3.8	98.37±0.1	98.00±0.0	98.49±0.1	97.97±0.2	97.96±0.5	97.76±0.5	97.72±0.4	**98.59±0.1**
Freebase	Ma-F1	20	45.14±4.5	53.81±0.6	53.96±0.7	55.78±0.5	52.72±1.0	53.16±2.8	54.90±0.7	55.79±0.9	**59.23±0.7**
		40	44.88±4.1	52.44±2.3	57.80±1.1	59.28±0.6	48.57±0.5	59.63±2.3	53.40±1.4	49.88±1.9	**61.19±0.6**
		60	45.16±3.1	50.65±0.4	55.94±0.7	56.50±0.4	52.37±0.8	56.77±1.7	53.81±1.1	52.10±0.7	**60.13±1.3**
	Mi-F1	20	54.83±3.0	55.20±0.7	56.23±0.8	57.92±0.5	56.85±0.9	57.24±3.2	58.16±0.9	58.26±0.9	**61.72±0.6**
		40	57.08±3.2	56.05±2.0	61.01±1.3	62.71±0.7	53.96±1.1	63.74±2.7	57.82±0.8	54.28±1.6	**64.03±0.7**
		60	55.92±3.2	53.85±0.4	58.74±0.8	58.57±0.5	56.84±0.7	61.06±2.0	57.96±0.7	56.69±1.2	**63.61±1.6**
	AUC	20	67.63±5.0	73.03±0.7	71.78±0.7	73.89±0.4	70.84±0.7	73.26±2.1	72.80±0.6	73.19±1.2	**76.22±0.8**
		40	66.42±4.7	74.05±0.9	75.51±0.8	76.08±0.4	69.48±0.2	77.74±1.2	72.97±1.1	70.77±1.6	**78.44±0.5**
		60	66.78±3.5	71.75±0.4	74.78±0.4	74.89±0.4	71.01±0.5	75.69±1.5	73.32±0.9	73.17±1.4	**78.04±0.4**
AMiner	Ma-F1	20	42.46±2.5	60.22±2.0	54.78±0.5	58.32±1.1	50.06±0.9	56.07±3.2	51.61±3.2	59.50±2.1	**71.38±1.1**
		40	45.77±1.5	65.66±1.5	64.77±0.5	64.50±0.7	58.97±0.9	63.85±1.5	54.72±2.6	61.92±2.1	**73.75±0.5**
		60	44.91±2.0	63.74±1.6	60.65±0.3	65.53±0.7	57.34±1.4	62.02±1.2	55.45±2.4	61.15±2.5	**75.80±1.8**
	Mi-F1	20	49.68±3.1	65.78±2.9	60.82±0.4	63.64±1.1	61.49±2.5	68.86±4.6	62.39±3.9	63.93±3.3	**78.81±1.3**
		40	52.10±2.2	71.34±1.8	69.66±0.6	71.57±0.7	68.47±2.2	76.89±1.6	63.87±2.9	63.60±2.5	**80.53±0.7**
		60	51.36±2.2	67.70±1.9	63.92±0.5	69.76±0.8	65.61±2.2	74.73±1.4	63.10±3.0	62.51±2.6	**82.46±1.4**
	AUC	20	70.86±2.5	85.39±1.0	81.22±0.3	83.35±0.5	77.96±1.4	78.92±2.3	75.89±2.2	85.34±0.9	**90.82±0.6**
		40	74.44±1.3	88.29±1.0	88.82±0.2	88.70±0.4	83.14±1.6	80.72±2.1	77.86±2.1	88.02±1.3	**92.11±0.6**
		60	74.16±1.3	86.92±0.8	85.57±0.2	87.74±0.5	84.77±0.9	80.39±1.5	77.21±1.4	86.20±1.7	**92.40±0.7**

Table 3: Quantitative results (%±σ) on node clustering.

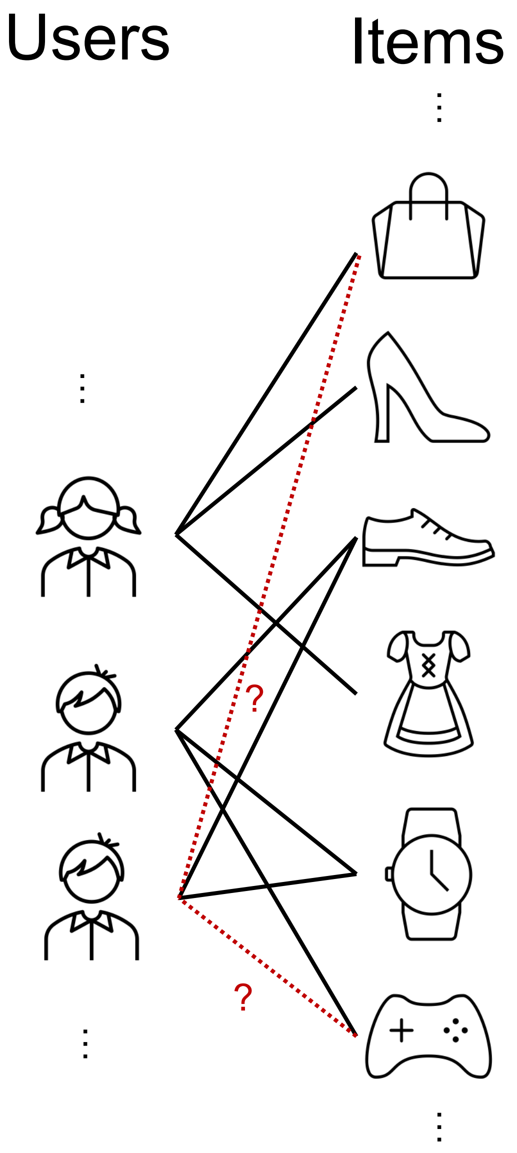
Datasets	ACM		DBLP		Freebase		AMiner	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
GraphSage	29.20	27.72	51.50	36.40	9.05	10.49	15.74	10.10
GAE	27.42	24.49	72.59	77.31	19.03	14.10	28.58	20.90
Mp2vec	48.43	34.65	73.55	77.70	16.47	17.32	30.80	25.26
HERec	47.54	35.67	70.21	73.99	19.76	19.36	27.82	20.16
HetGNN	41.53	34.81	69.79	75.34	12.25	15.01	21.46	26.60
DGI	51.73	41.16	59.23	61.85	18.34	11.29	22.06	15.93
DMGI	51.66	46.64	70.06	75.46	16.98	16.91	19.24	20.09
HeCo	**56.87**	**56.94**	**74.51**	**80.17**	**20.38**	**20.98**	**32.26**	**28.64**

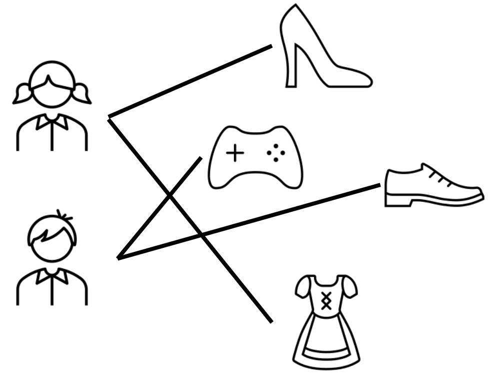
Wang, X., Liu, N., Han, H., & Shi, C. (2021, August). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1726-1736).

Part 4: Applications of graph self-supervised learning

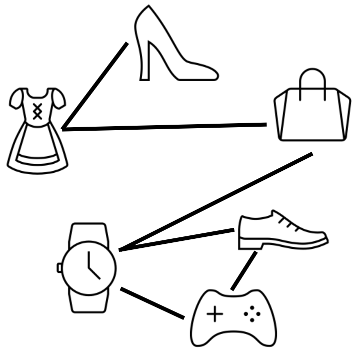
- Recommender system
- Outlier detection
- More applications: Chemistry, graph structure learning…

Graphs in recommender system

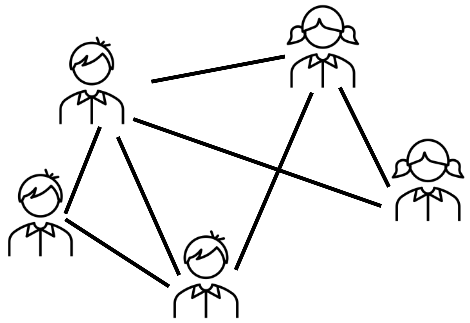




Interaction → User-item graph



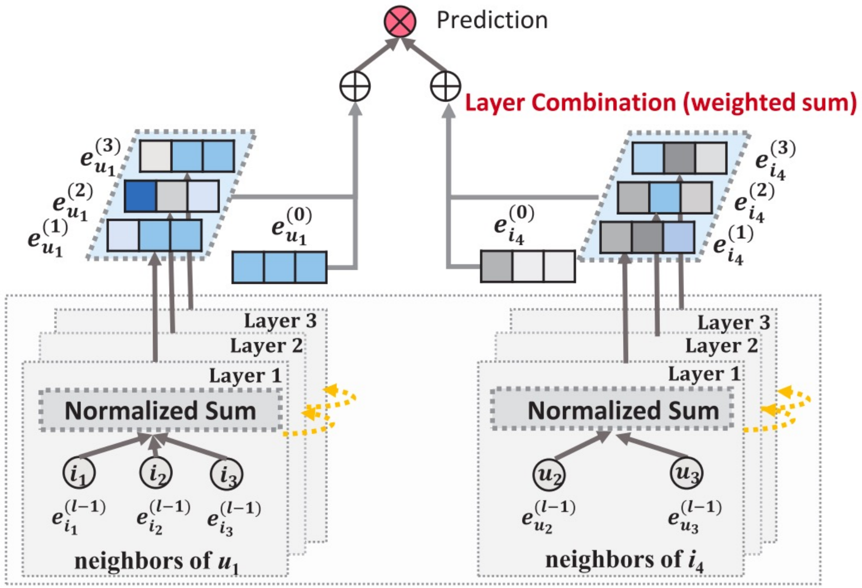
Item transmission/similarity→ Item-item graph
(for sequential/session-based recommendation)



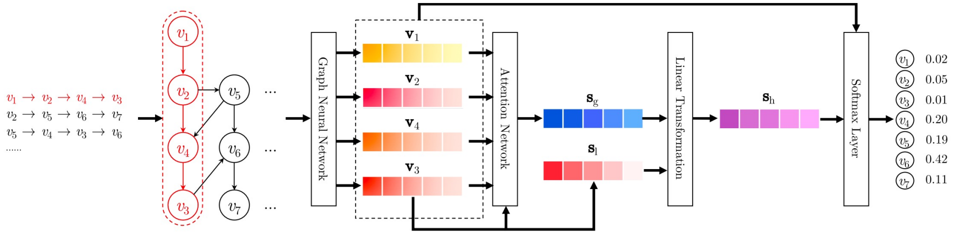
Social relation → User-user graph
(for social recommendation)

…

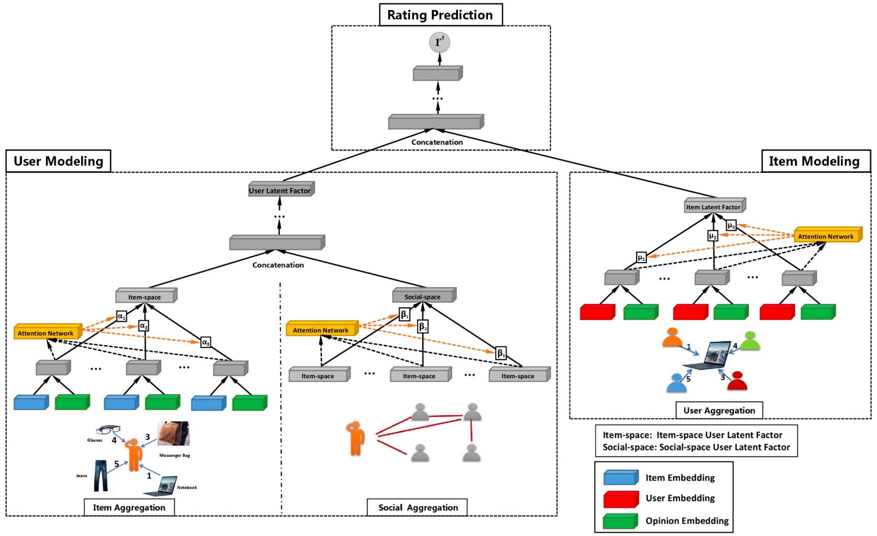
GNNs for recommender system



LightGCN for collaborative filtering



SR-GNN for session-based recommendation



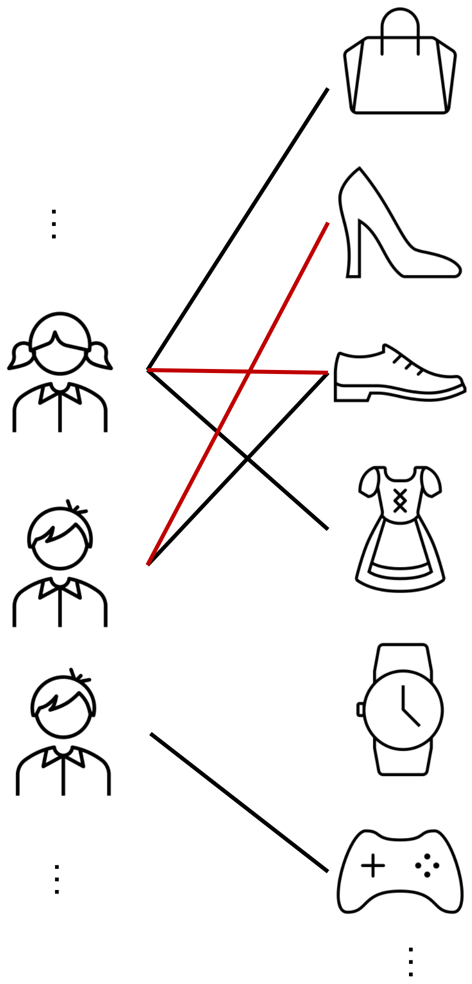
GraphRec for social recommendation

...

...

[1] He, Xiangnan, et al. "Lightgcn: Simplifying and powering graph convolution network for recommendation." Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.
[2] Wu, Shu, et al. "Session-based recommendation with graph neural networks." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.
[3] Fan, Wenqi, et al. "Graph neural networks for social recommendation." The world wide web conference. 2019.

GSSL for recommendation: Motivations



Learning scheme: observed interactions → ranking loss (e.g. BPR)

$$\mathcal{L}_{main} = \sum_{(u,i,j)\in O} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}),$$

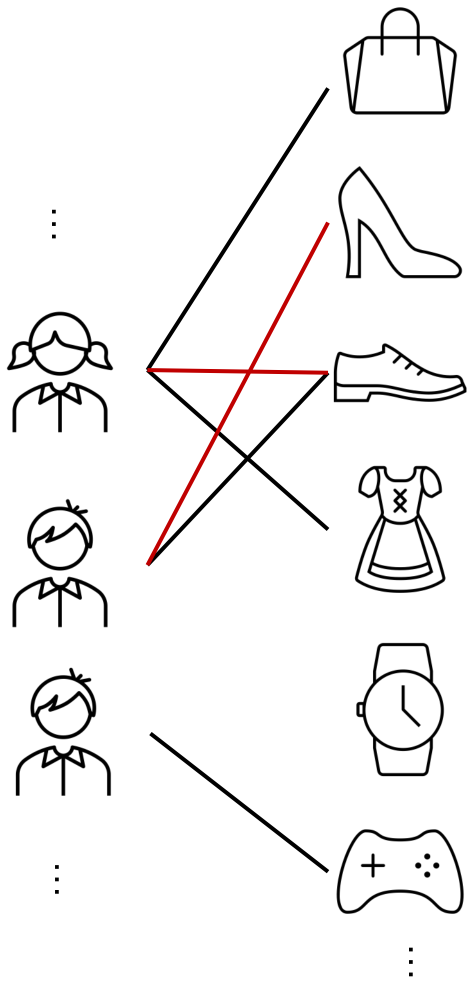
- Problem 1: Sparse Supervision Signal

The observed interactions can be extremely sparse compared to the whole interaction space

GSSL:
provide extra supervision signals from data itself!

Wu, Jiancan, et al. "Self-supervised graph learning for recommendation." Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021.

GSSL for recommendation: Motivations



Learning scheme: observed interactions → ranking loss (e.g. BPR)

$$\mathcal{L}_{main} = \sum_{(u,i,j)\in O} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}),$$

- Problem 1: Sparse Supervision Signal

- Problem 2: Noisy interaction

Observed interactions usually contain noises, e.g., a user is misled to click an item and finds it uninteresting after consuming it

GSSL:

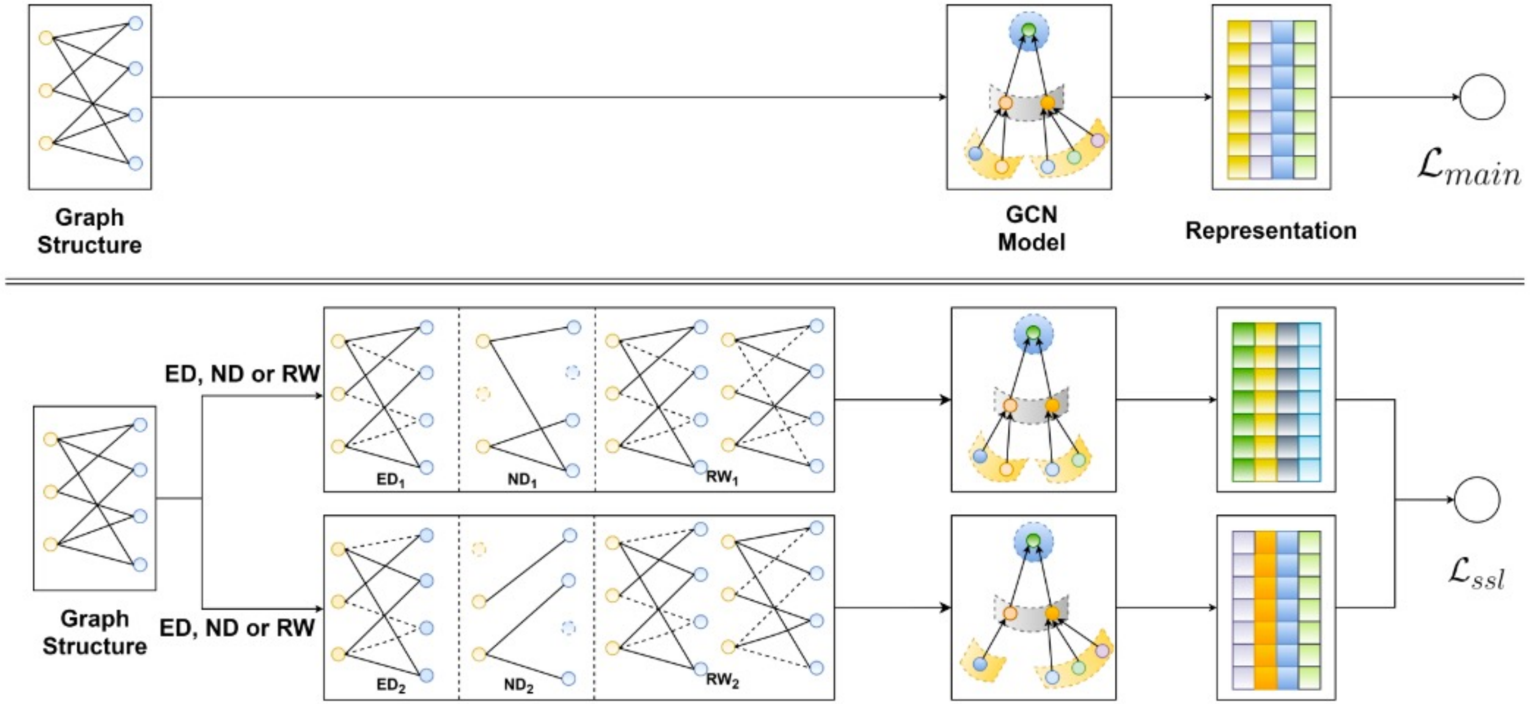
- **Regularize the model to prevent it from over-fitting the noisy interaction**
- **Data augmentations to reduce the impact by noise**

Wu, Jiancan, et al. "Self-supervised graph learning for recommendation." Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021.

Contrast-based method

SGL

Scenario: collaborative filtering recommendation



$$\mathcal{L}_{ssl}^{user} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{z}'_u, \mathbf{z}''_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{z}'_u, \mathbf{z}''_v)/\tau)},$$

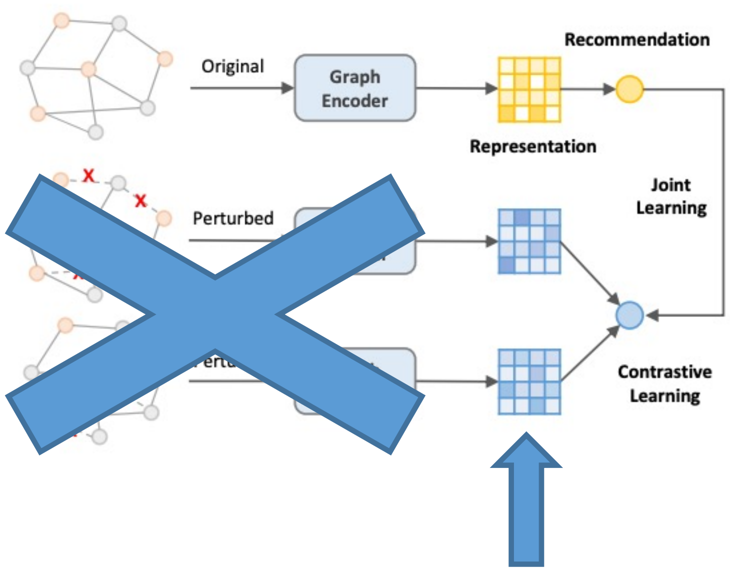
Augmentations: Node Dropout (ND), Edge Dropout (ED), and Random Walk (RW)

Wu, Jiancan, et al. "Self-supervised graph learning for recommendation." Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021.

Contrast-based method

Following works of SGL: Scenario: collaborative filtering recommendation

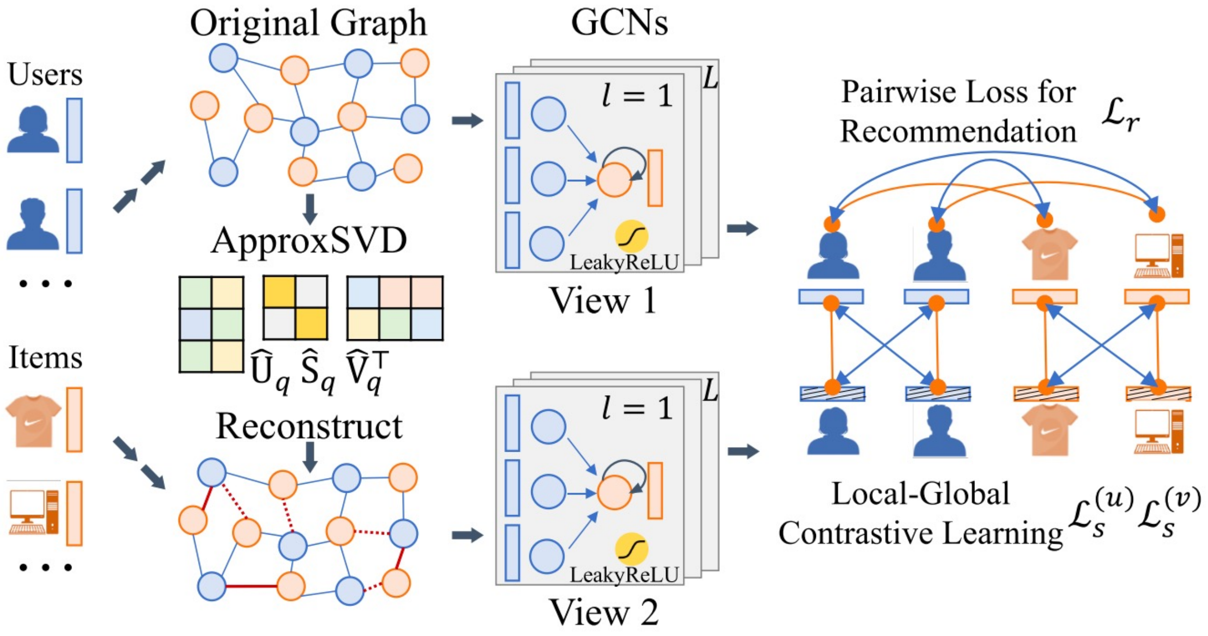
SimGCL



$$\mathbf{e}_i' = \mathbf{e}_i + \Delta_i', \quad \mathbf{e}_i'' = \mathbf{e}_i + \Delta_i'',$$

representation-level augmentation!

LightGCL



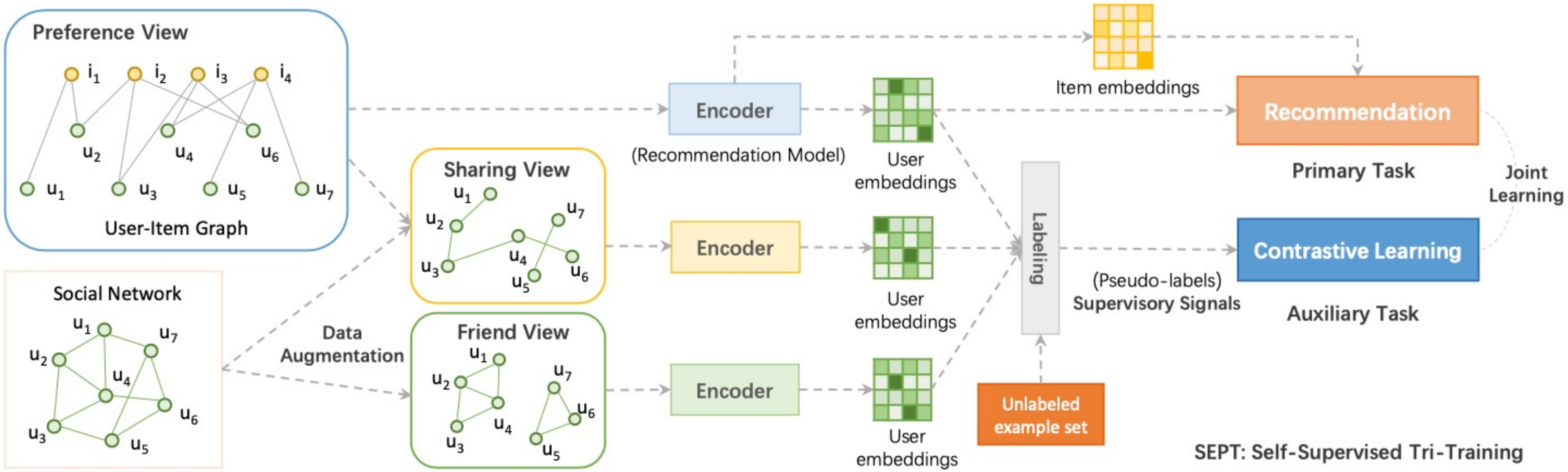
SVD-based augmentation

Yu, Junliang, et al. "Are graph augmentations necessary? simple graph contrastive learning for recommendation." Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022.
Cai, Xuheng, et al. "LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation." ICLR 2023

Contrast-based method

SEPT

Scenario: social recommendation

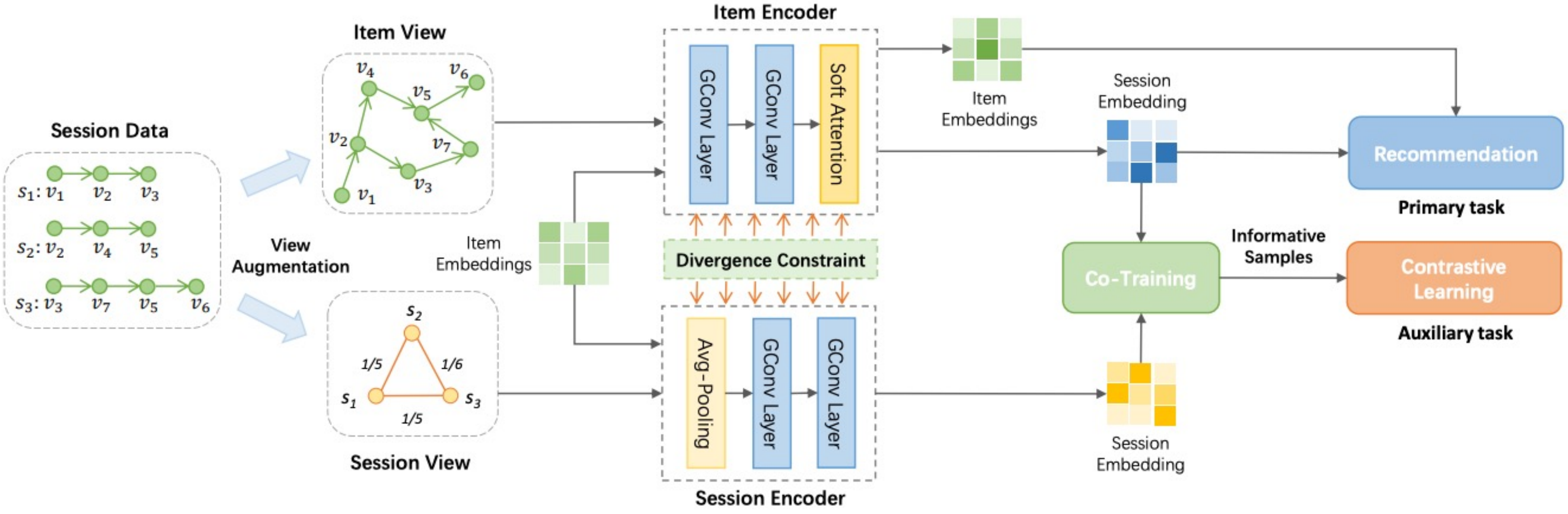


Yu, Junliang, et al. "Socially-aware self-supervised tri-training for recommendation." Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021.

Contrast-based method

COTREC

Scenario: session-based recommendation

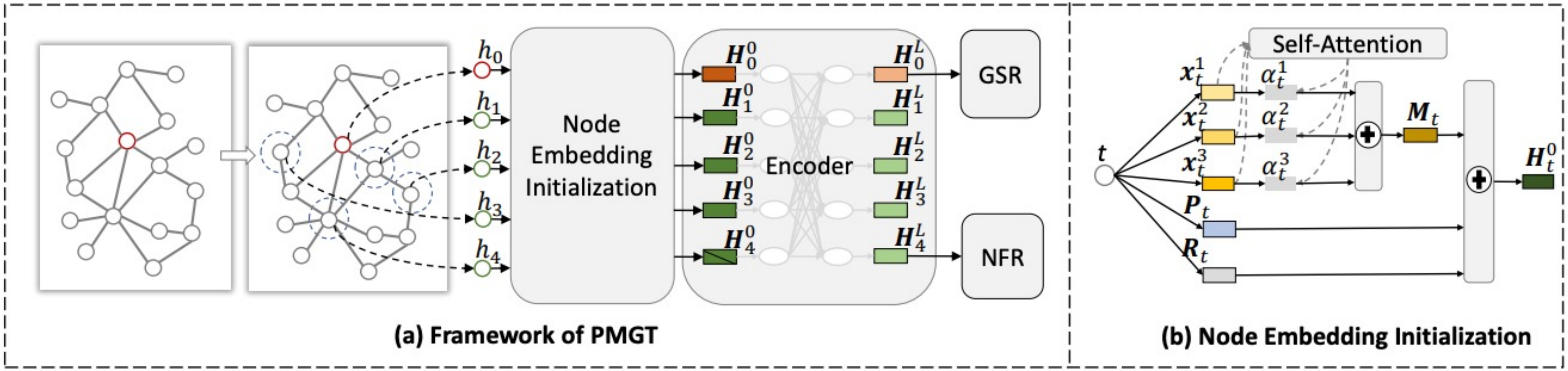


Xia, Xin, et al. "Self-supervised graph co-training for session-based recommendation." Proceedings of the 30th ACM International conference on information & knowledge management. 2021.

Generation-based method

PMGT

Scenario: Multimodal Side Information-based Recommendation



Loss:

Edge reconstruction:

$$\mathcal{L}_{edge} = \frac{1}{|\mathcal{V}|} \sum_{h \in \mathcal{V}} \frac{1}{|\mathcal{N}_h|} \sum_{t \in \mathcal{N}_h} \Big[-\log\big(\sigma(\frac{\mathbf{h}^\top \mathbf{t}}{||\mathbf{h}||_2 ||\mathbf{t}||_2})\big) - Q \cdot \mathbb{E}_{t_n \sim P_n(t)} \log\big(\sigma(-\frac{\mathbf{h}^\top \mathbf{t}_n}{||\mathbf{h}||_2 ||\mathbf{t}_n||_2})\big) \Big],$$

Feature reconstruction:

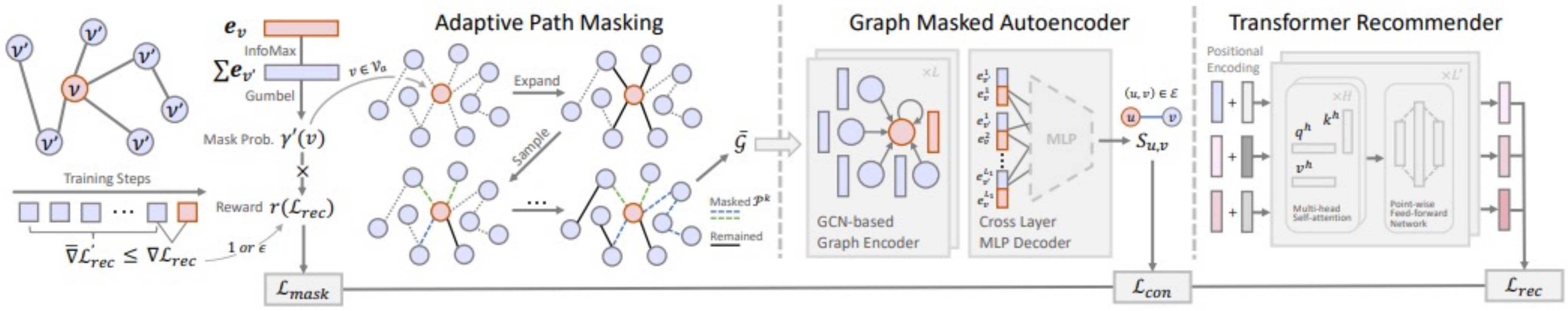
$$\mathcal{L}_{feature} = \frac{1}{|\mathcal{V}|} \sum_{h \in \mathcal{V}} \frac{1}{|\mathcal{M}_h|} \sum_{t \in \mathcal{M}_h} \sum_{i}^{m} \|\mathbf{H}_t^L \mathbf{W}_r^i - \mathbf{x}_t^i\|_2^2,$$

Liu, Yong, et al. "Pre-training graph transformer with multimodal side information for recommendation." Proceedings of the 29th ACM International Conference on Multimedia, 2021

Generation-based method

MAERec

Scenario: Sequential Recommendation



$$\gamma'(v) = \gamma(v) - \log(-\log(\mu)), \quad \mu \sim \text{Uniform}(0, 1).$$

$$\mathcal{L}_{mask} = -\sum_{v \in \mathcal{V}} \gamma'(v)$$

"Learning to mask" loss

$$\mathcal{L}_{con} = -\sum_{(v,v') \in \mathcal{E} \setminus \mathcal{P}^k} \log \frac{\exp(s_{v,v'})}{\sum_{v'' \in \mathcal{V}} \exp(s_{v,v''})}$$

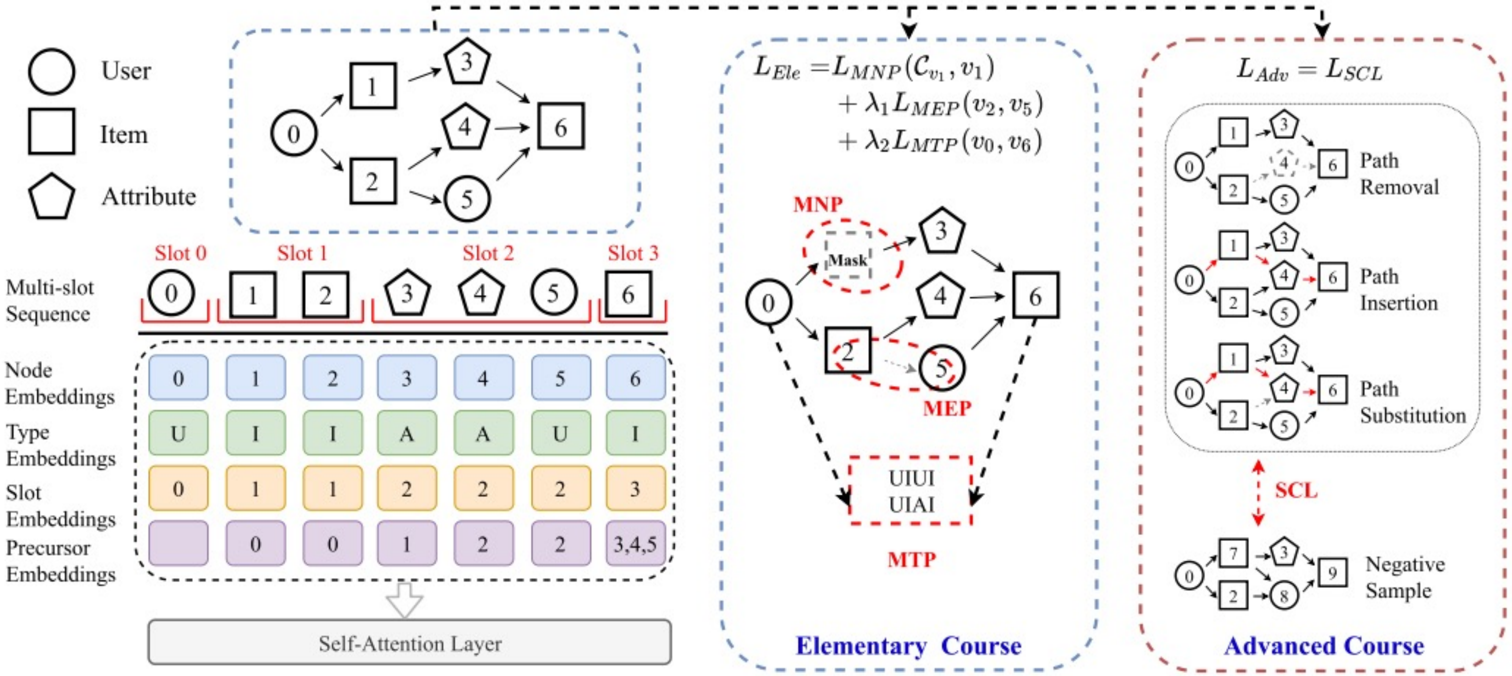
Reconstruction loss:
recovering the masked global item transition paths

Ye, Yaowen, Lianghao Xia, and Chao Huang. "Graph Masked Autoencoder for Sequential Recommendation." SIGIR 2023

Hybrid method

CHEST

Scenario: Sequential Recommendation



Three tasks:
- Masked Node Prediction (MNP)
- Masked Edge Prediction (MEP)
- Meta-path Type Prediction (MTP)

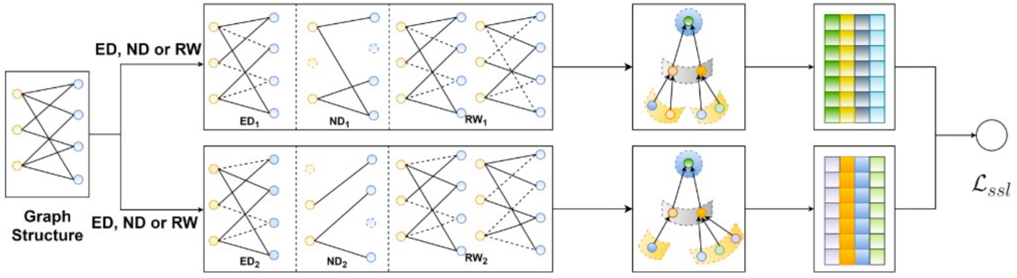
Wang, Hui, et al. "Curriculum Pre-Training Heterogeneous Subgraph Transformer for Top-N Recommendation." ACM Transactions on Information Systems 41.1 (2023): 1-28.

Summary: GSSL for recommender systems

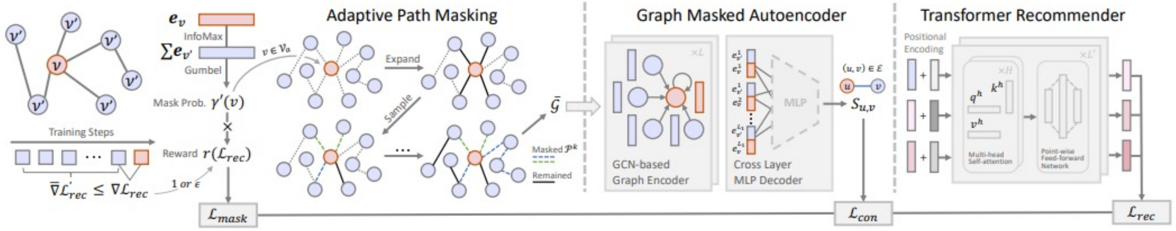
- **Scenarios**
  - Collaborative filtering-based recommendation
  - Social recommendation
  - Session-based recommendation
  - Sequential recommendation
  - …

- **Pretext tasks**
  - Mainstream solution: Contrast-based GSSL
  - Promising directions: Generation-based and hybrid GSSL

- **Representative methods**

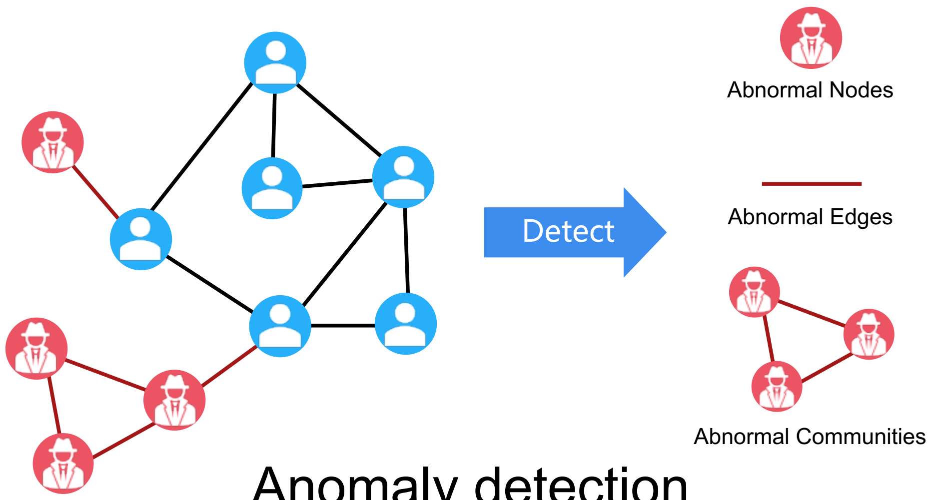


SGL

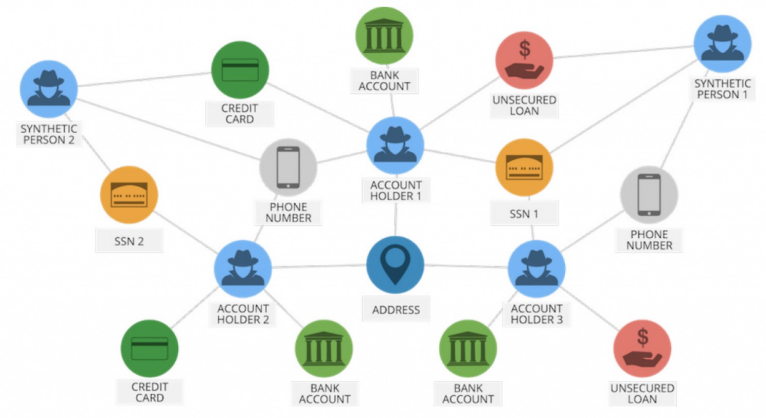


MAERec

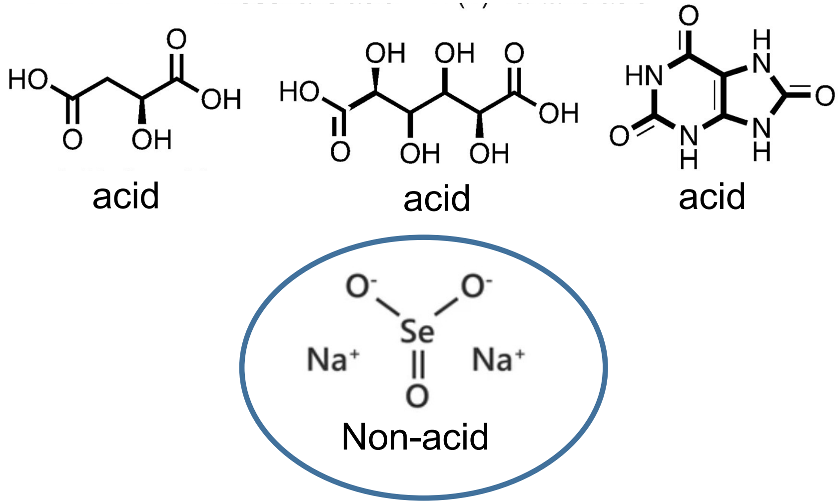
Graph-based outlier detection



Anomaly detection



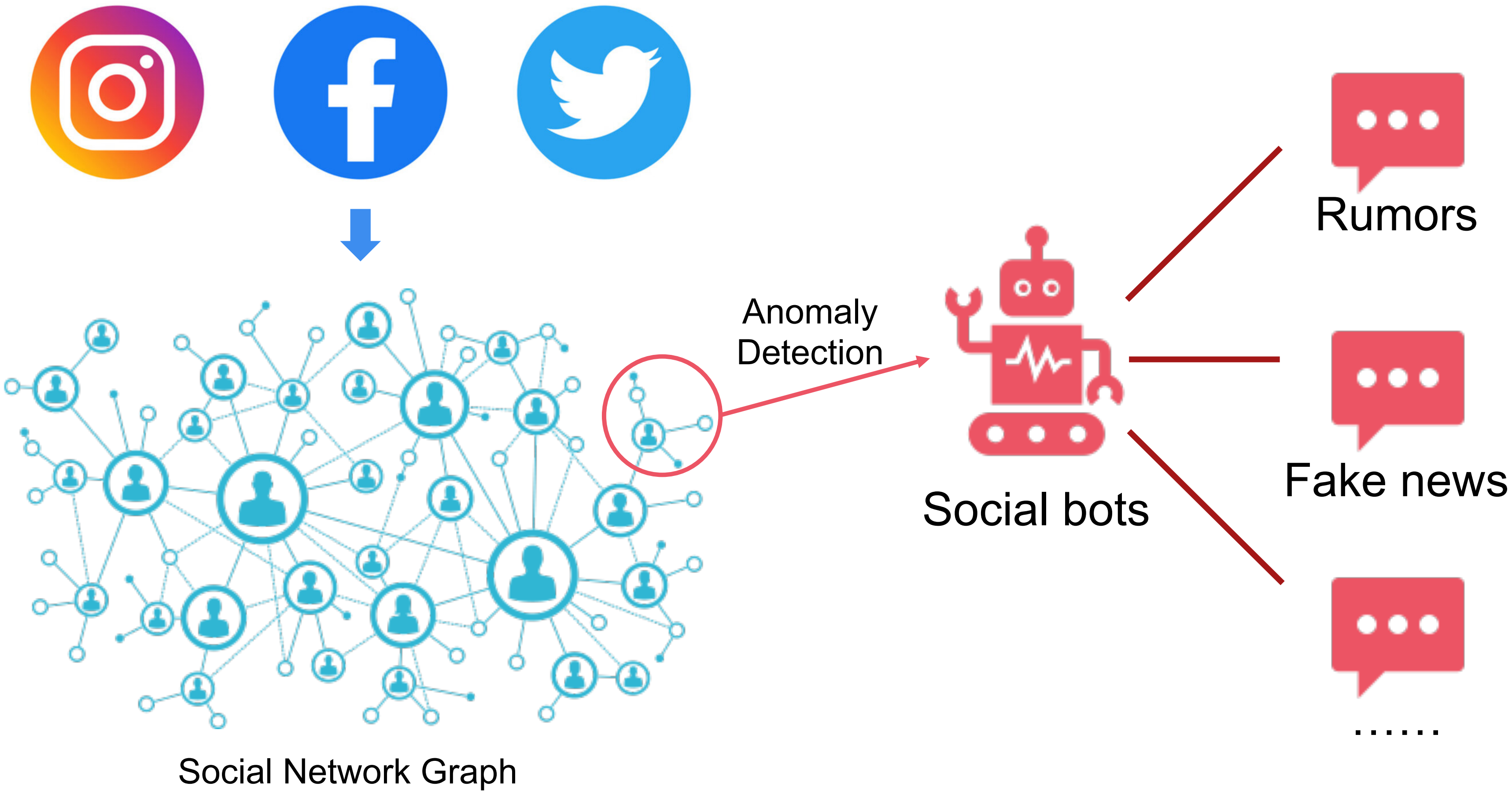
Fraud detection



Out-of-distribution detection

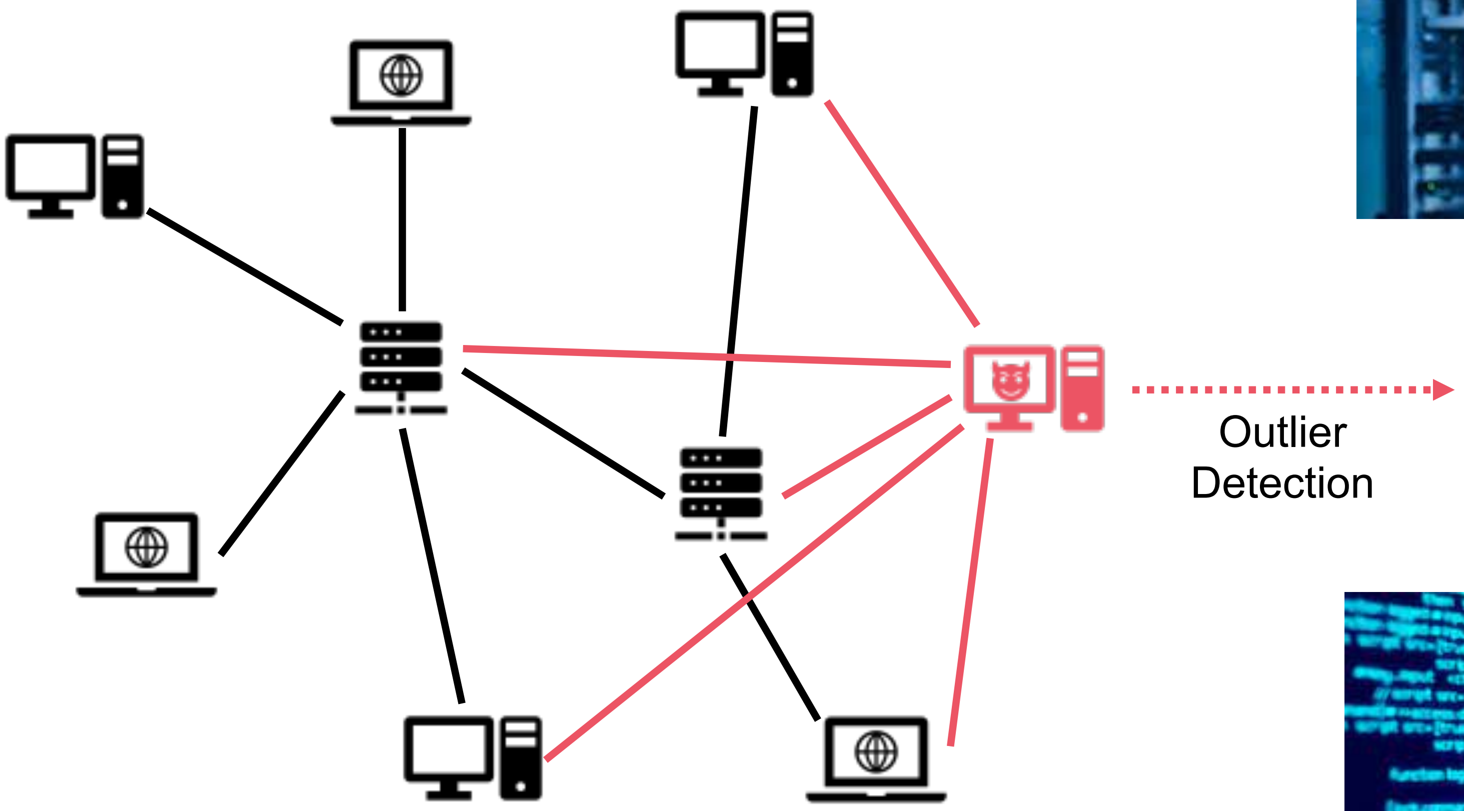
https://neo4j.com/blog/enterprise-fraud-detection/

Graph-based outlier detection



https://neo4j.com/blog/enterprise-fraud-detection/

Graph-based outlier detection



Outlier Detection

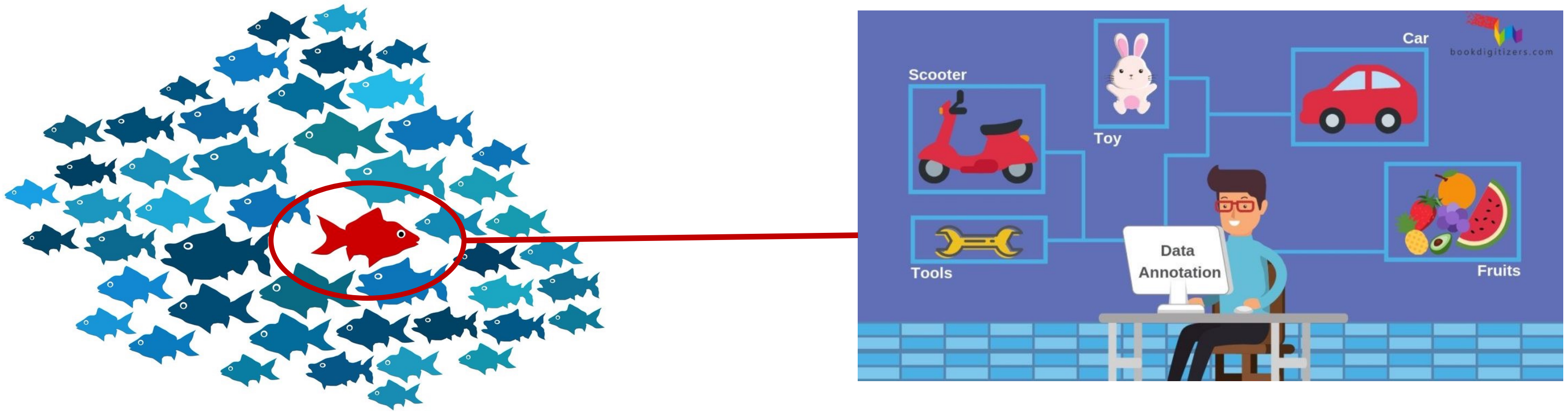
Hackers

Cyber Attacks

https://neo4j.com/blog/enterprise-fraud-detection/

GSSL for outlier detection: motivation

The lack of annotated labels for outliers:

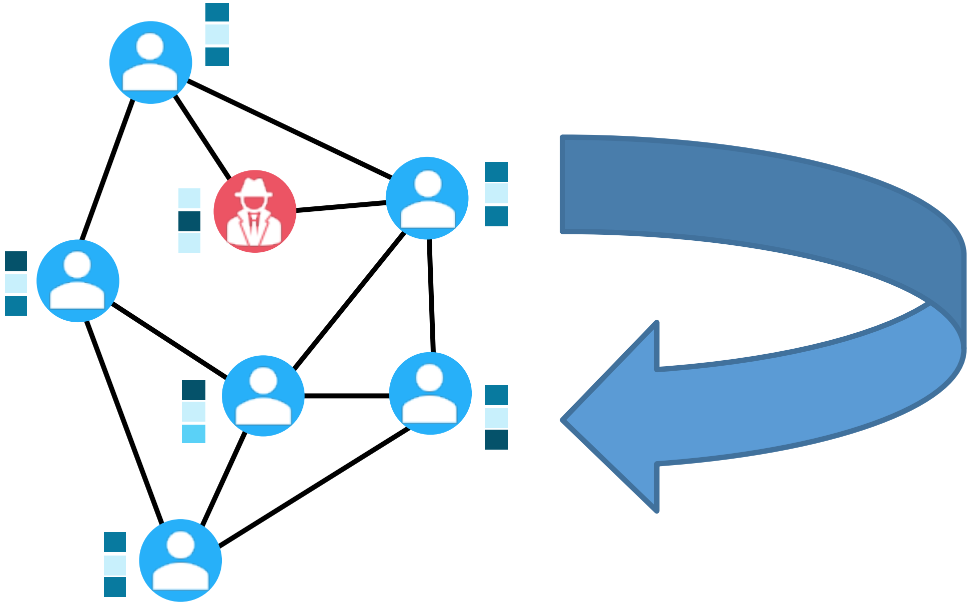


Challenge: It’s difficult to annotate the anomalies/out-of-distribution samples from numerous normal sample!

GSSL for outlier detection: motivation

It’s difficult to annotate the anomalies/out-of-distribution samples from numerous normal sample!

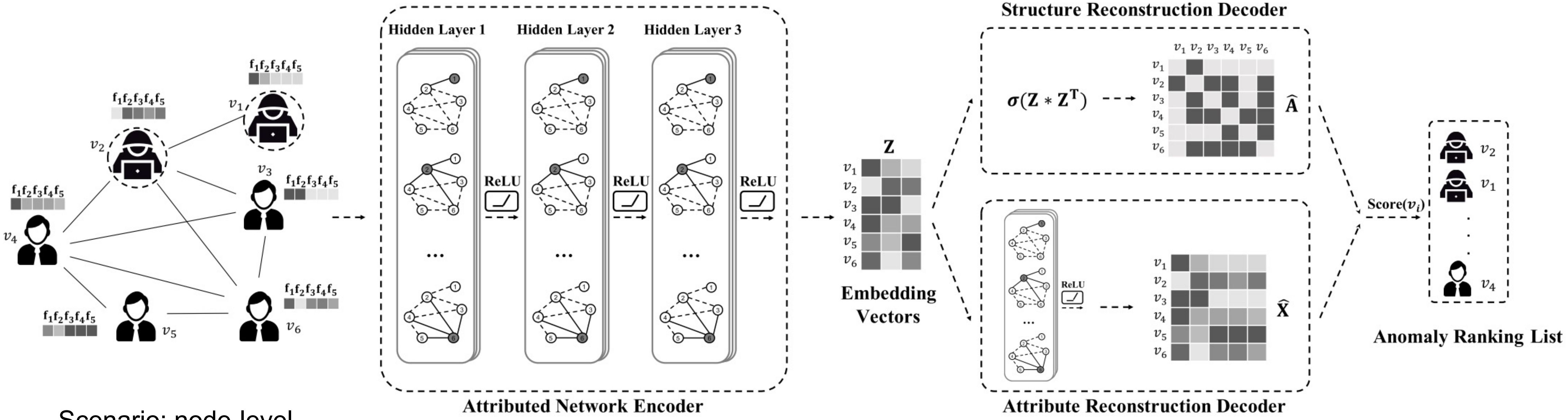
Self-supervised methods:
capture the latent patterns of normal data without any label
→ the model can find the outlier according to its normality



Capture the normal patterns from itself!

Generation-based method

DOMINANT

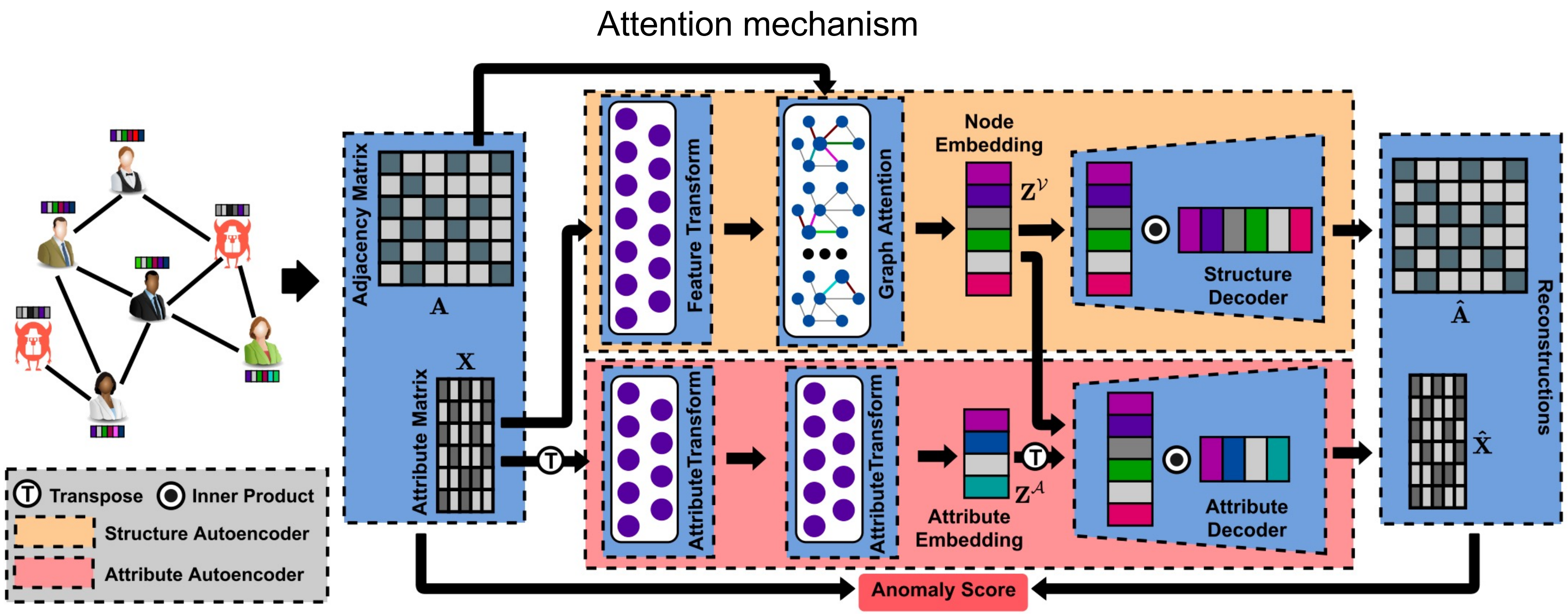


Scenario: node-level anomaly detection

Ding, Kaize, et al. "Deep anomaly detection on attributed networks." Proceedings of the 2019 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2019.

Generation-based method

AnomalyDAE



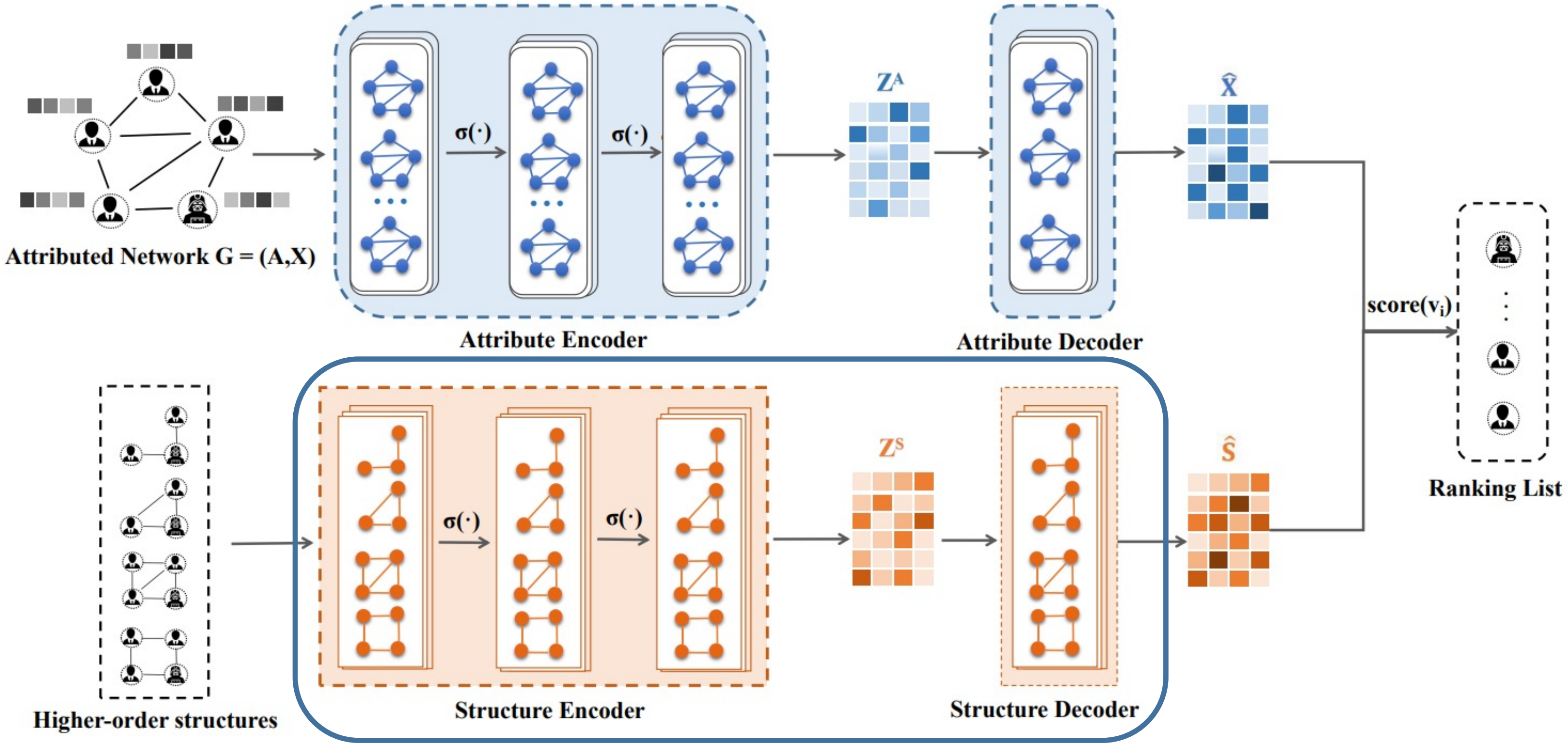
Scenario: node-level anomaly detection

Fan, Haoyi, Fengbin Zhang, and Zuoyong Li. "Anomalydae: Dual autoencoder for anomaly detection on attributed networks." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

Generation-based method

GUIDE

Scenario: node-level anomaly detection

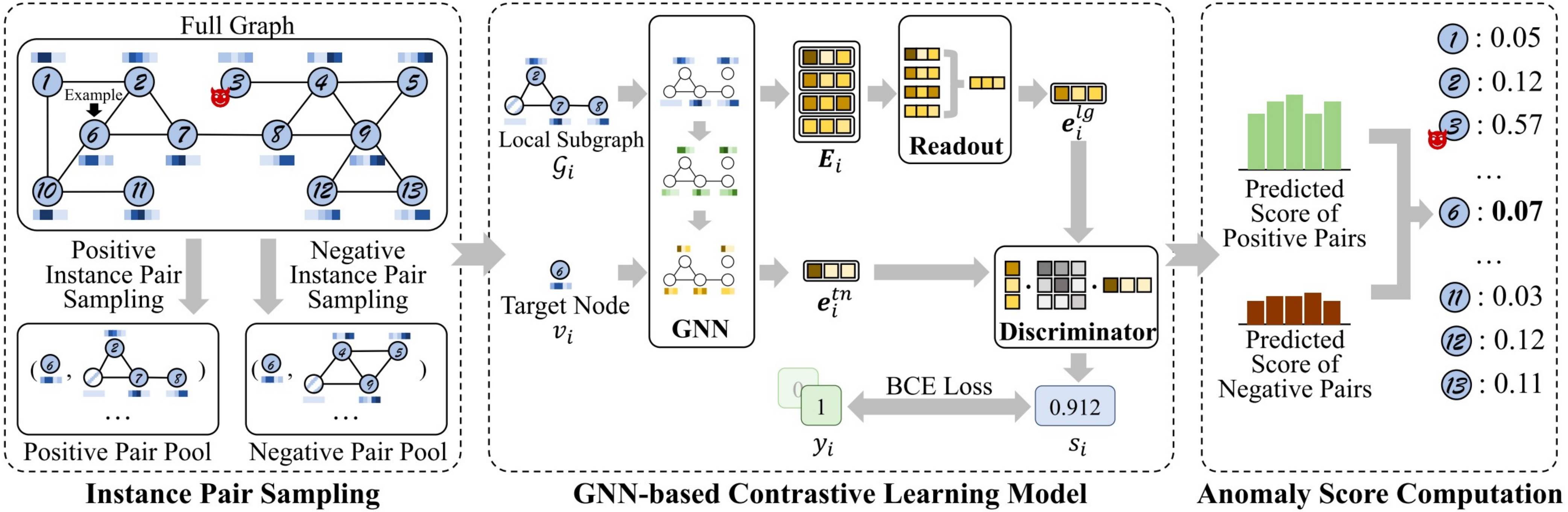


Consider various motifs in structure-based auto-encoder

Yuan, Xu, et al. "Higher-order structure based anomaly detection on attributed networks." 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021.

Contrast-based method

CoLA

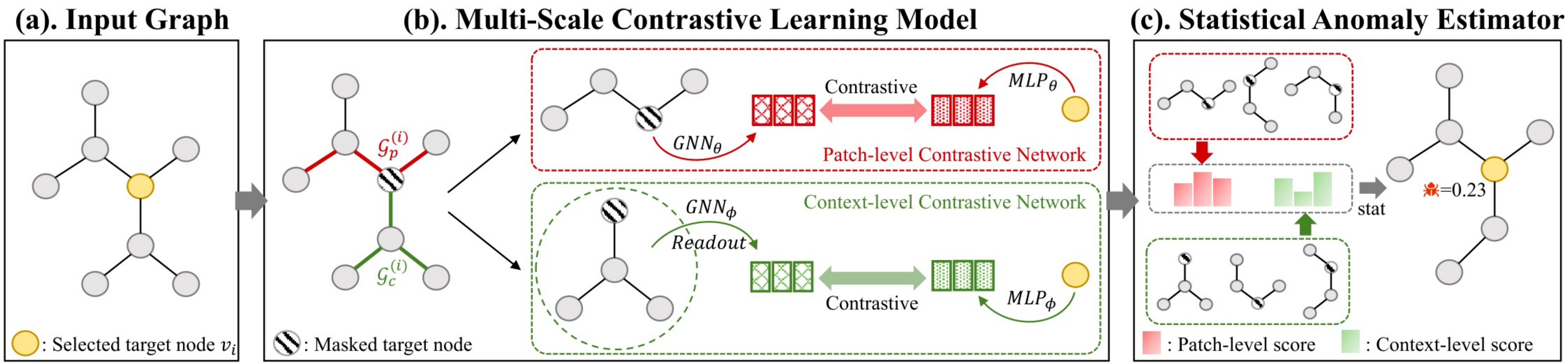


<u>Scenario: node-level anomaly detection</u>

Liu, Yixin, et al. "Anomaly detection on attributed networks via contrastive self-supervised learning." IEEE transactions on neural networks and learning systems 33.6 (2021): 2378-2392.

Contrast-based method

ANEMONE



Scenario: node-level anomaly detection

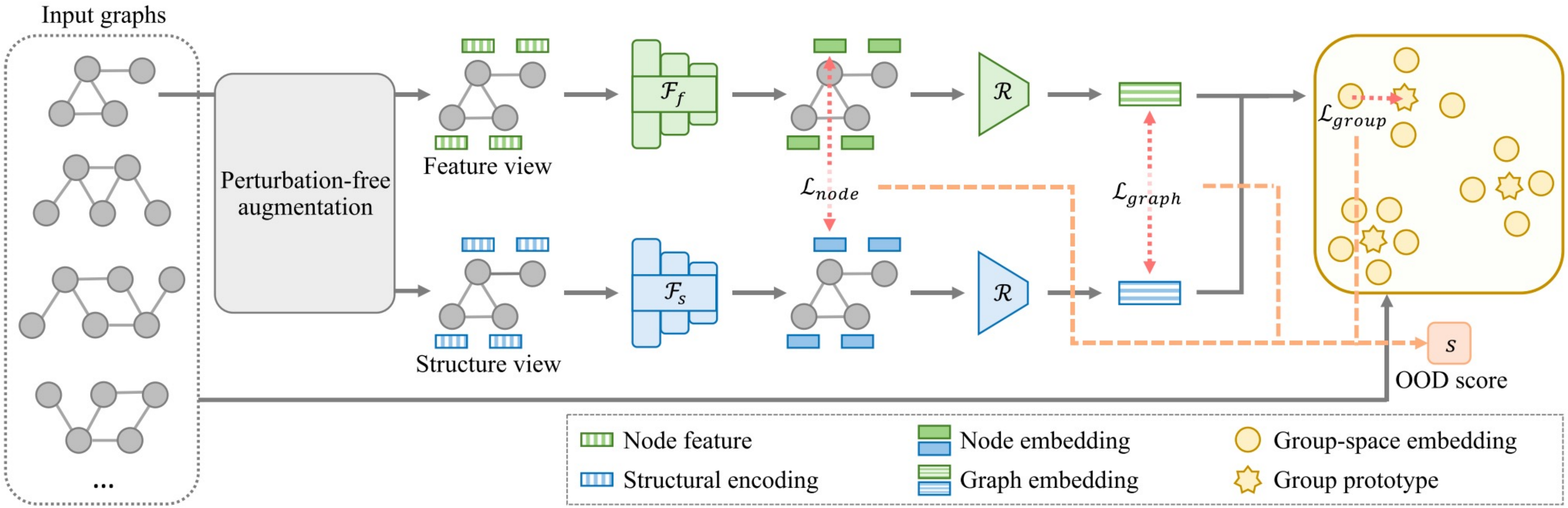
Multi-scale contrastive learning!

Jin, Ming, et al. "Anemone: Graph anomaly detection with multi-scale contrastive learning." Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.

Contrast-based method

GOOD-D

Hierarchical contrastive learning



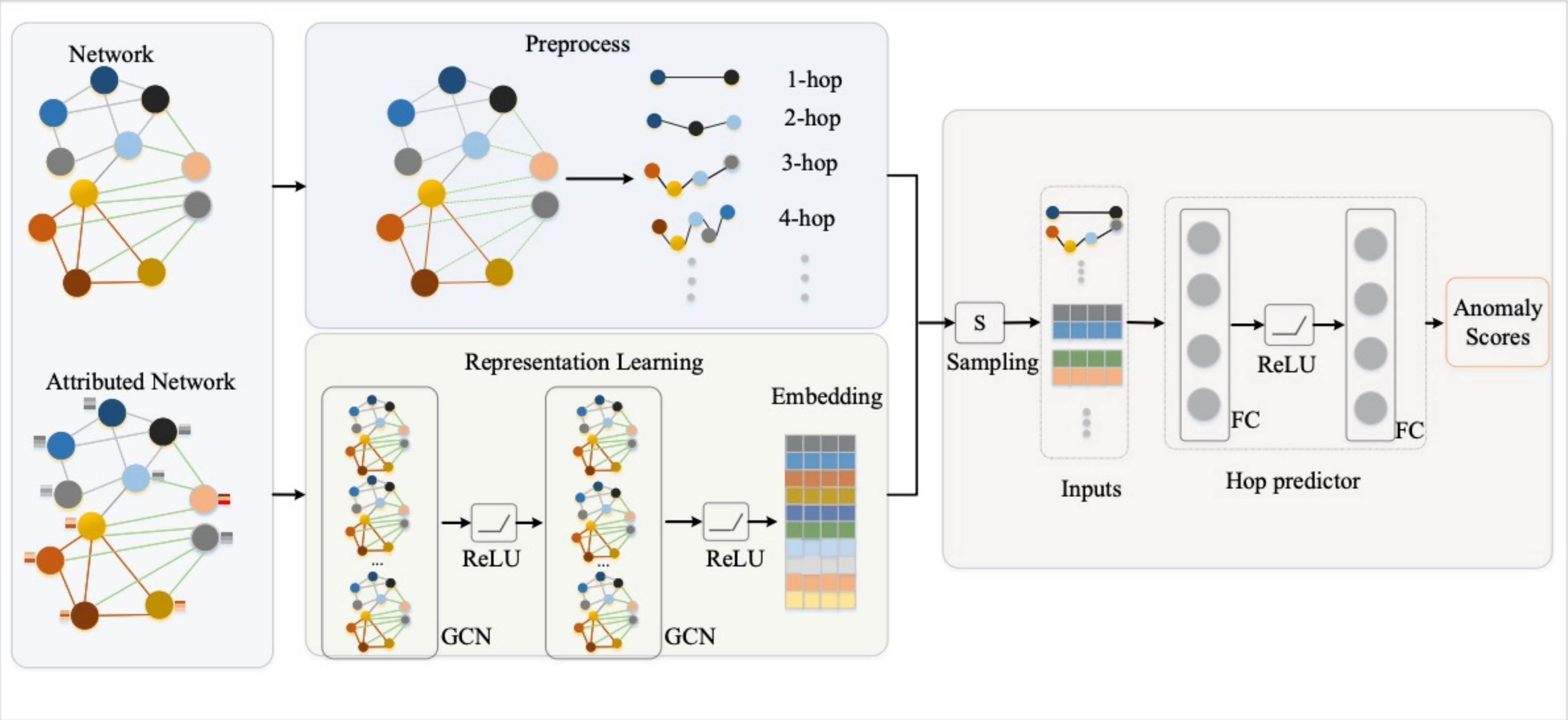
<u>Scenario: graph-level out-of-distribution/anomaly detection</u>

Liu, Yixin, et al. "GOOD-D: On Unsupervised Graph Out-Of-Distribution Detection." Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 2023.

Auxiliary property-based method

Sub-CR

Hop prediction-based anomaly detection



Scenario: node-level anomaly detection

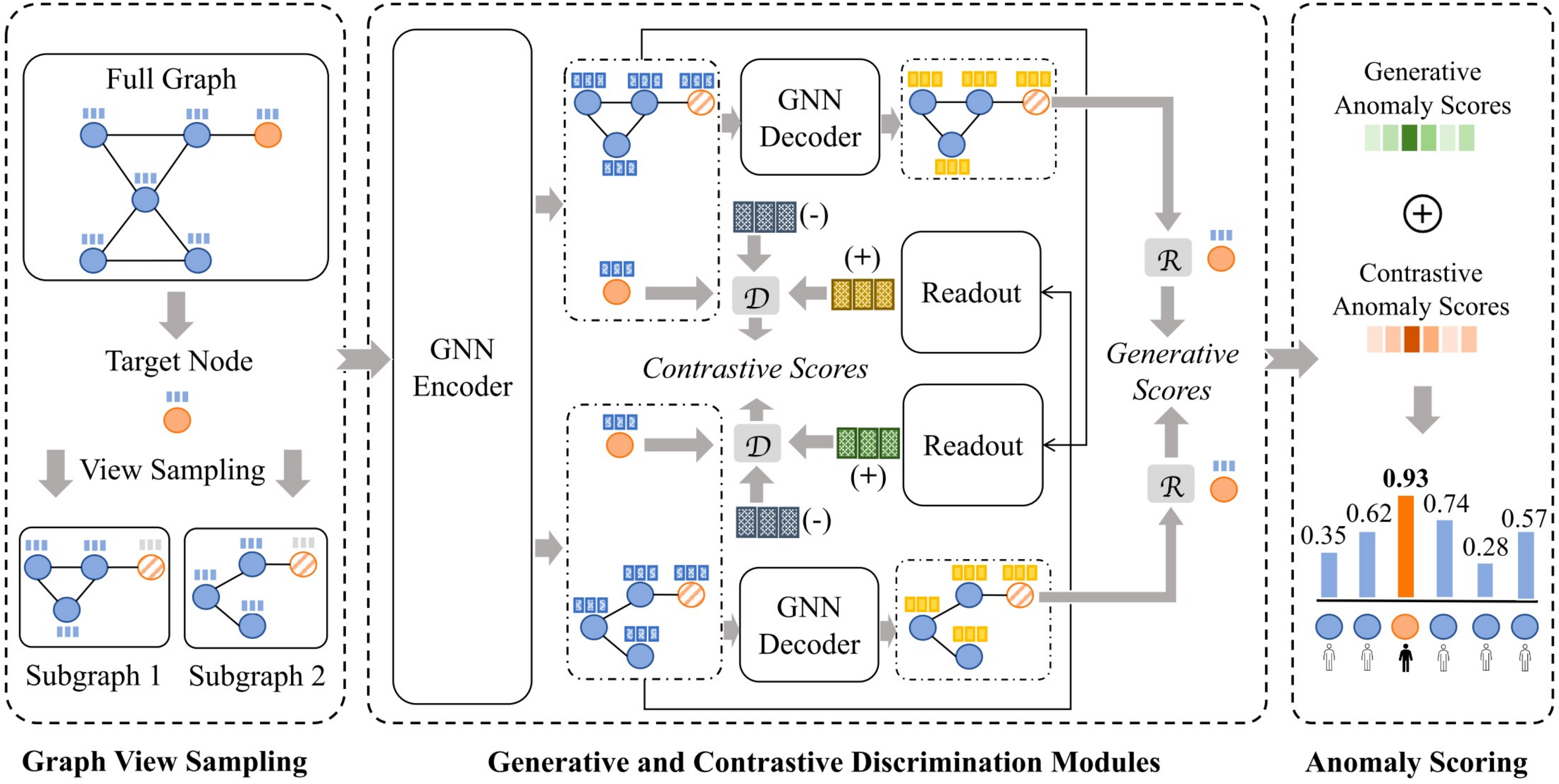
Huang, Tianjin, et al. "Hop-count based self-supervised anomaly detection on attributed networks." Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I. Cham: Springer International Publishing, 2023.

Hybrid method

SL-GAD

Scenario: node-level anomaly detection

Contrast-based + generation-based

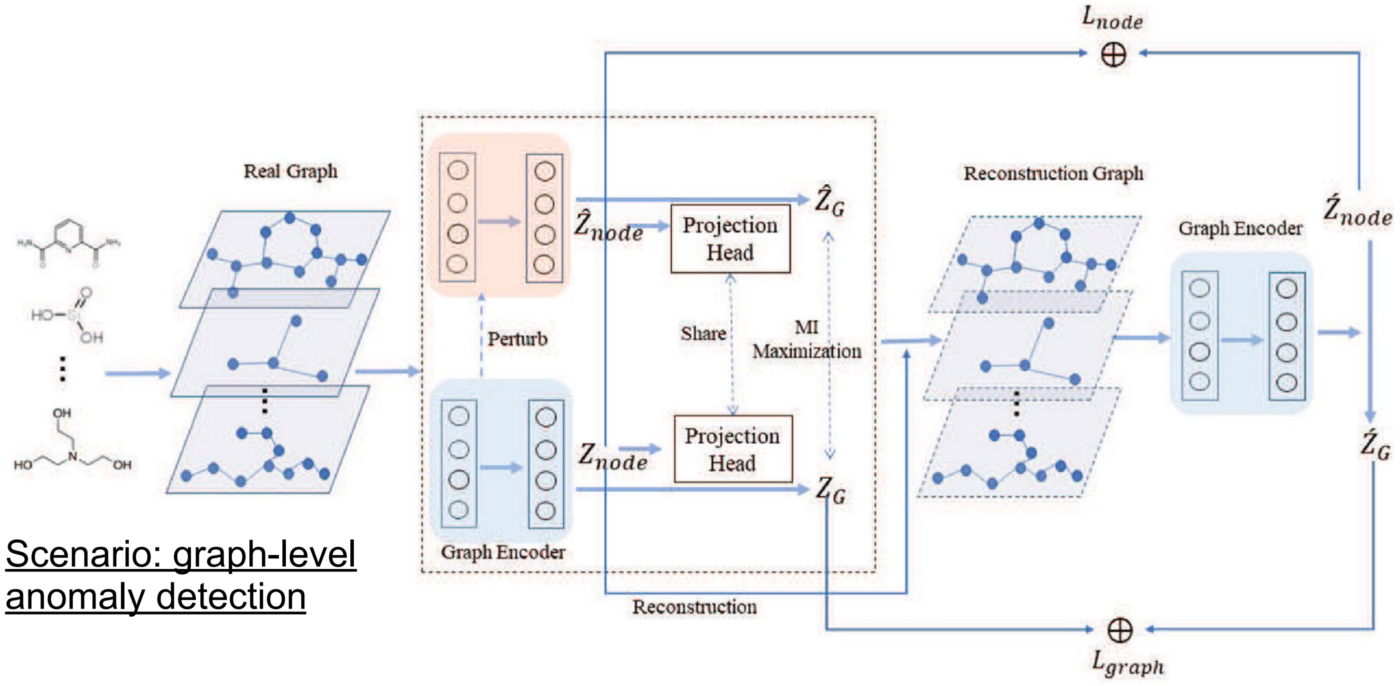


Zheng, Yu, et al. "Generative and contrastive self-supervised learning for graph anomaly detection." IEEE Transactions on Knowledge and Data Engineering (2021).

Hybrid method

GLADC

Contrast-based + generation-based



Scenario: graph-level anomaly detection

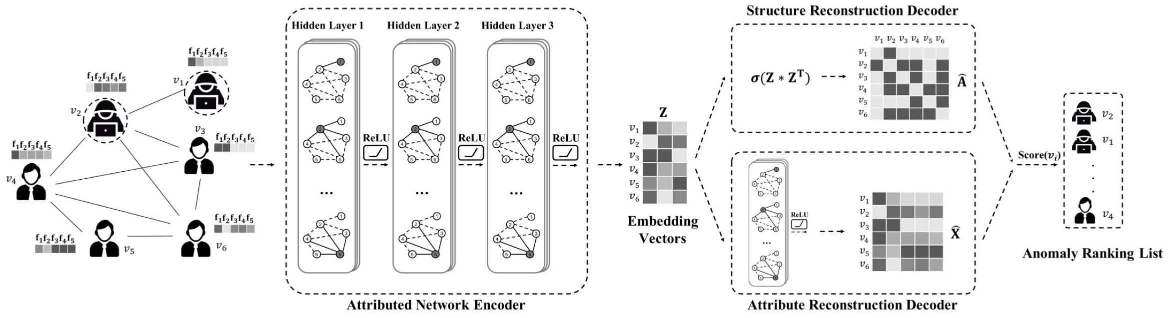
Luo, Xuexiong, et al. "Deep graph level anomaly detection with contrastive learning." Scientific Reports 12.1 (2022): 19867.

Summary: GSSL for outlier detection

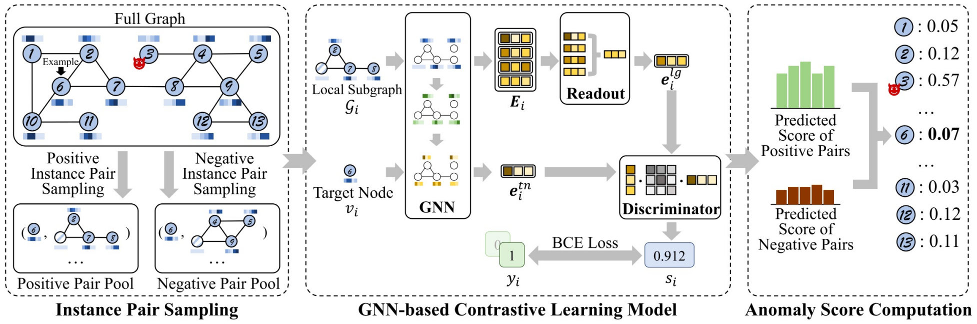
- **Scenarios**
  - Node-level
  - Graph-level

- **Pretext tasks**
  - Early methods: generation-based: autoencoder
  - Mainstream methods: contrast-based: from single scale to multi-scale
  - A new perspective: auxiliary property – predict the hop
  - Advanced solutions: hybrid GSSL

- **Representative methods**



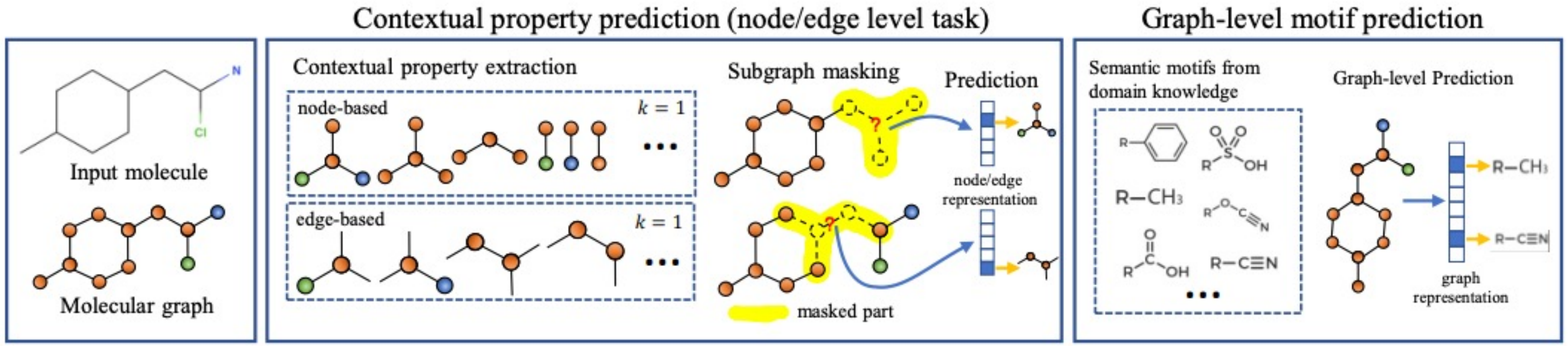
DOMINANT



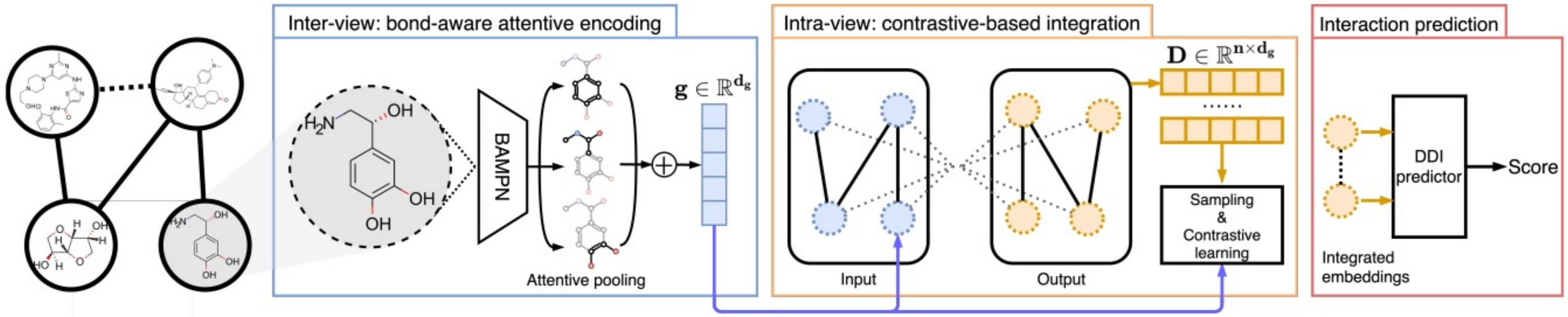
CoLA

More applications: chemistry

GROVER for molecular pre-train model



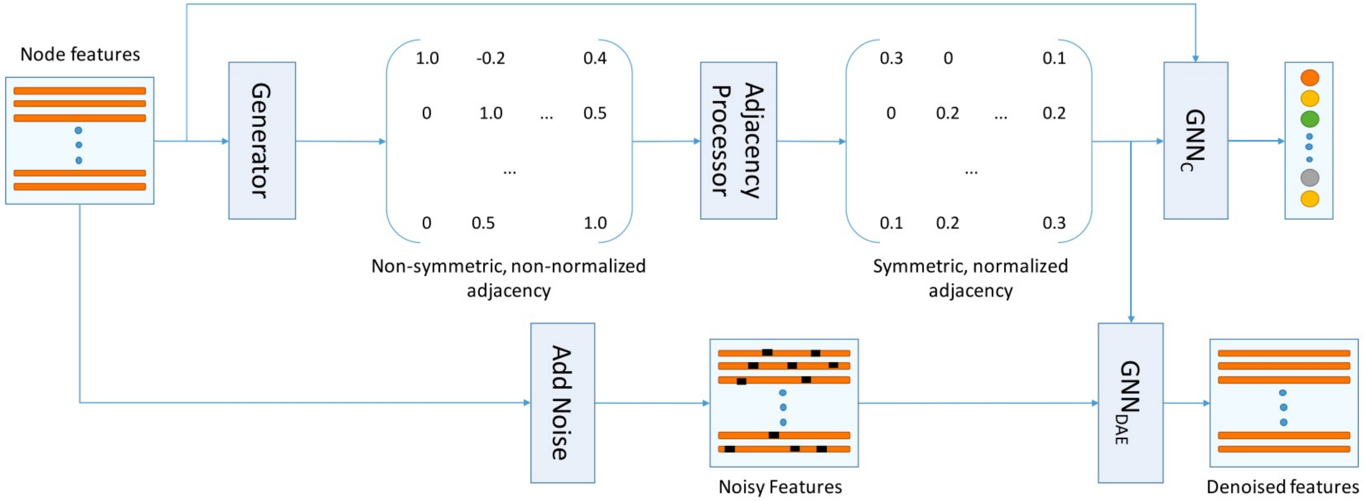
MIRACLE for drug-drug interaction prediction



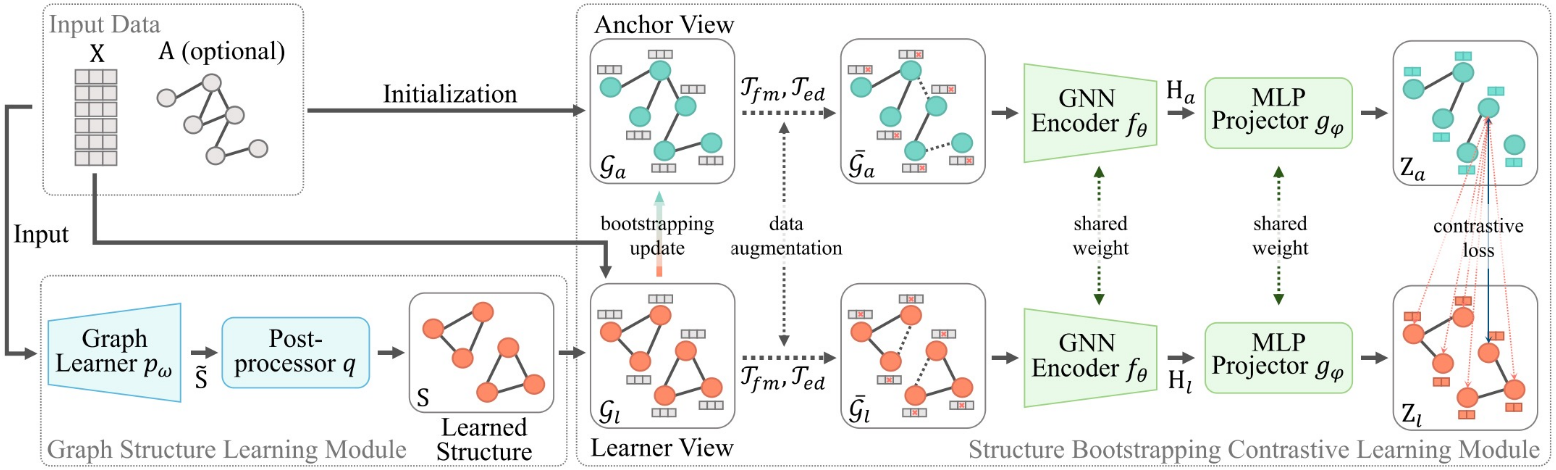
Rong, Yu, et al. "Self-supervised graph transformer on large-scale molecular data." *Advances in Neural Information Processing Systems* 33 (2020): 12559-12571.
Wang, Yingheng, et al. "Multi-view graph contrastive representation learning for drug-drug interaction prediction." Proceedings of the Web Conference 2021. 2021.

More applications: graph structure learning

SLAPS



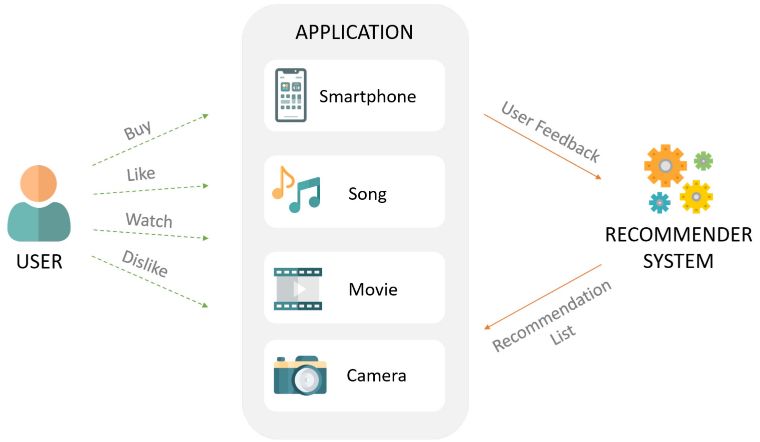
SUBLIME



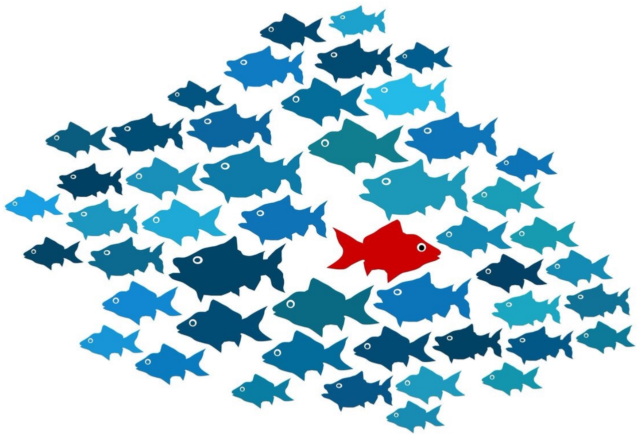
Fatemi, Bahare, Layla El Asri, and Seyed Mehran Kazemi. "SLAPS: Self-supervision improves structure learning for graph neural networks." Advances in Neural Information Processing Systems 34 (2021): 22667-22681.
Liu, Yixin, et al. "Towards unsupervised deep graph structure learning." Proceedings of the ACM Web Conference 2022. 2022.

Summary

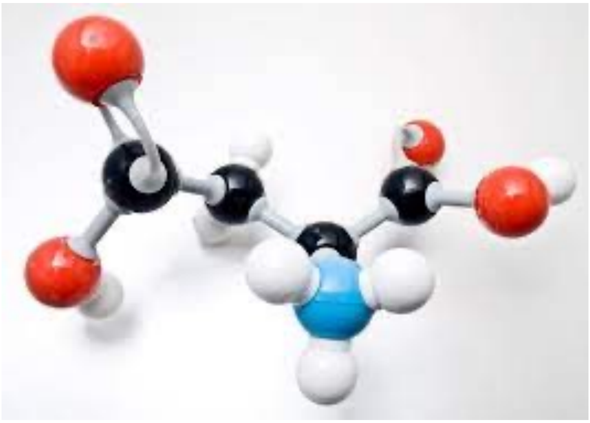
- **Recommender Systems**



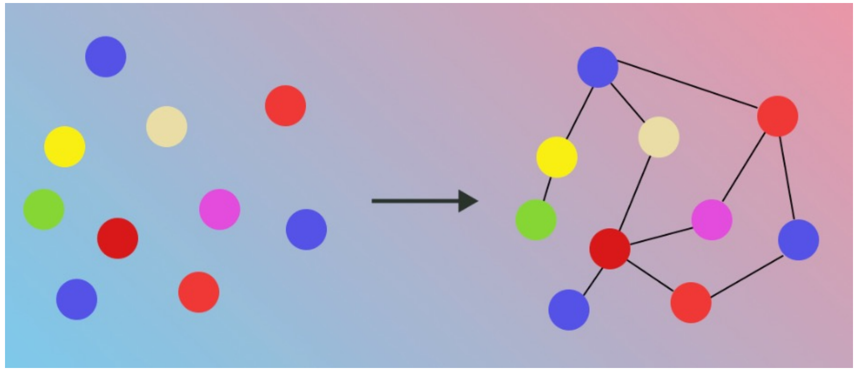
- **Outlier Detection**



- **Chemistry**



- **Graph Structure Learning**



- **Boarder Applications**

➢ Expert finding
➢ Program repairing
➢ Open world modeling
➢ Medical
➢ Federated Learning
…

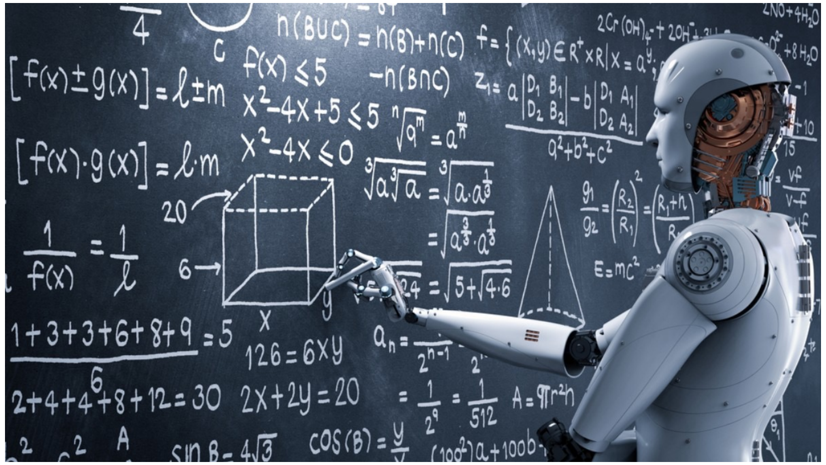
Part 5: Future directions and conclusion

- Potential directions of graph self-supervised learning
- Conclusion

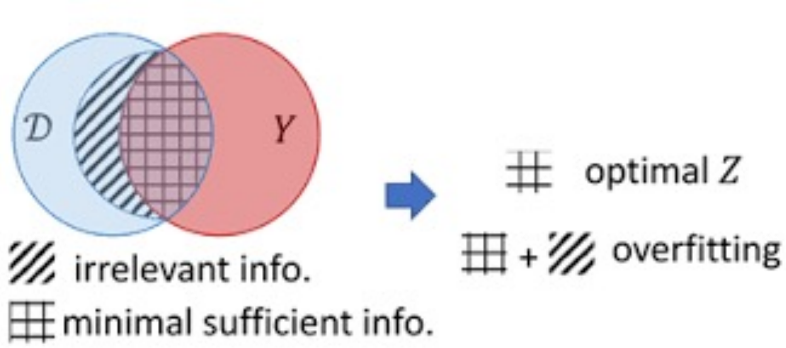
Future Directions

- Theoretical Foundation

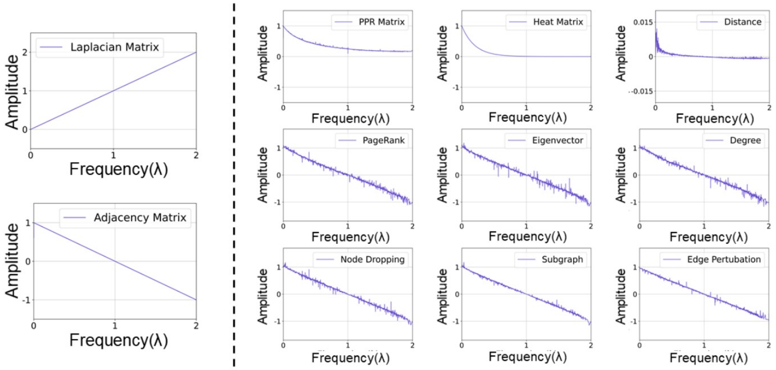
The existing methods are mostly designed with intuition and their performance gain is evaluated by empirical experiments, but don't have a solid theoretical foundation.



Potential theoretical basis:



Information theory



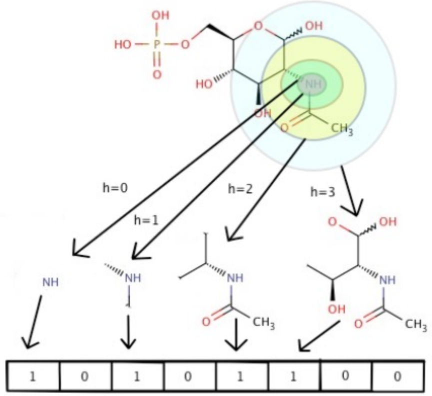
Spectral graph theory

…

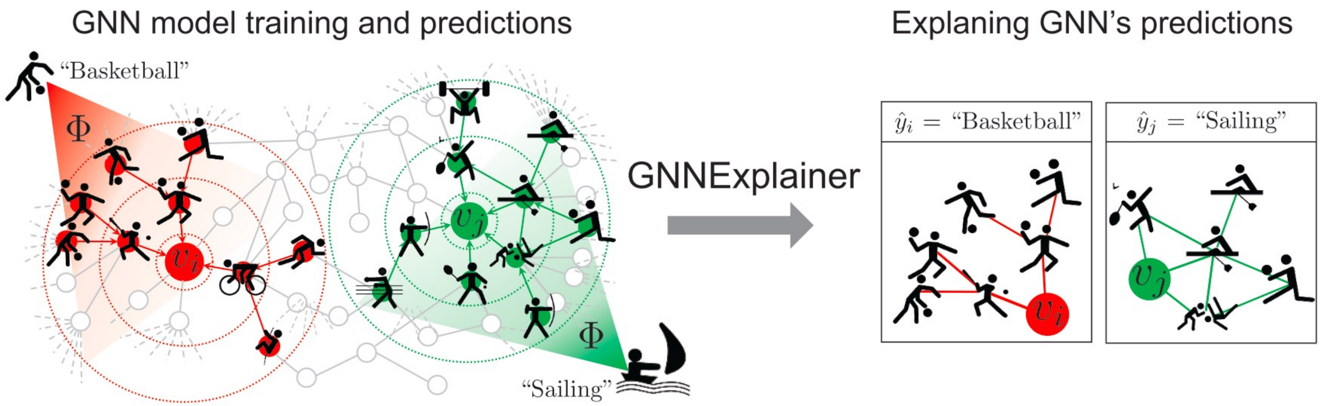
Wu, Tailin, et al. "Graph information bottleneck." Advances in Neural Information Processing Systems 33 (2020): 20437-20448.
Liu, Nian, et al. "Revisiting graph contrastive learning from the perspective of graph spectrum." Advances in Neural Information Processing Systems (2022)

Future Directions

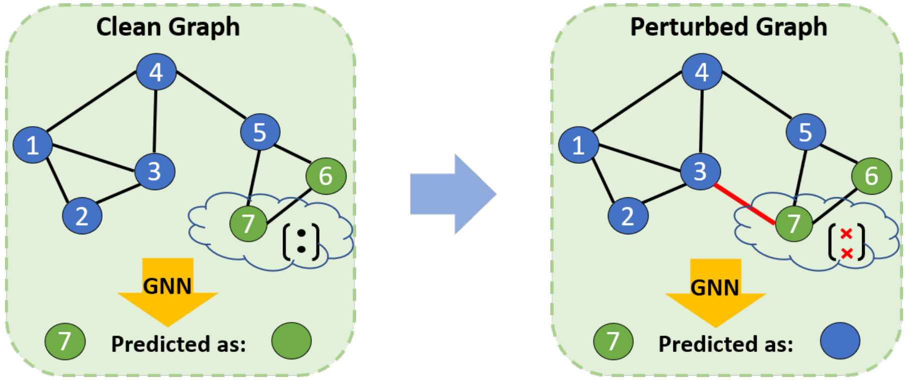
- Interpretability and Robustness



Most of the current works lack these properties.
Interpretability: Explainable GSSL model
Robustness: adversarial attack/defense of GSSL model



Interpretability



Adversarial Attack

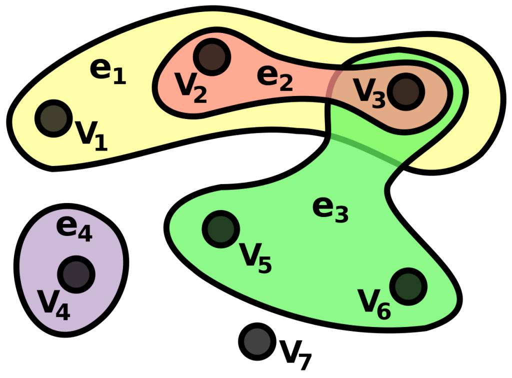
Ying, Zhitao, et al. "Gnnexplainer: Generating explanations for graph neural networks." Advances in neural information processing systems 32 (2019).
https://www.arxiv-vanity.com/papers/2003.00653/

Future Directions

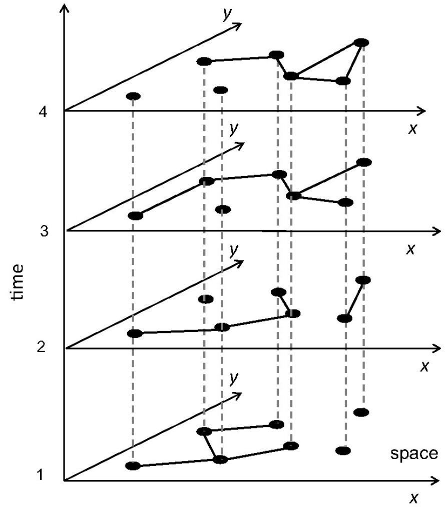
- Pretext Tasks for Complex Types of Graphs

Most of the existing works: Plain graph, Attributed graph

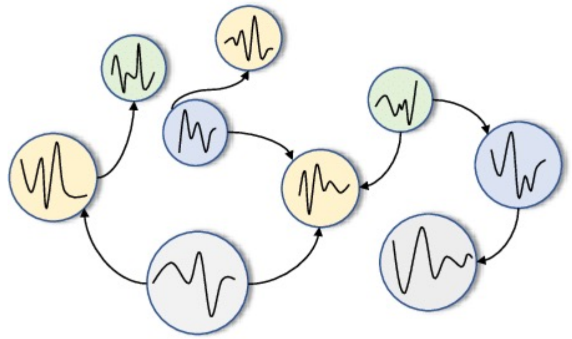
Potential targets:



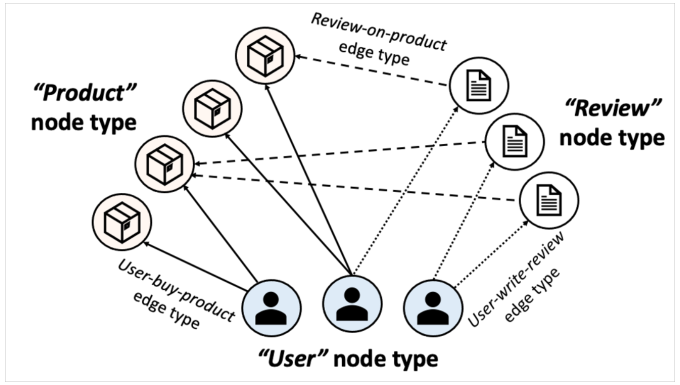
Hypergraph



Dynamic graph



Spatial-temporal graph



Heterogeneous graph

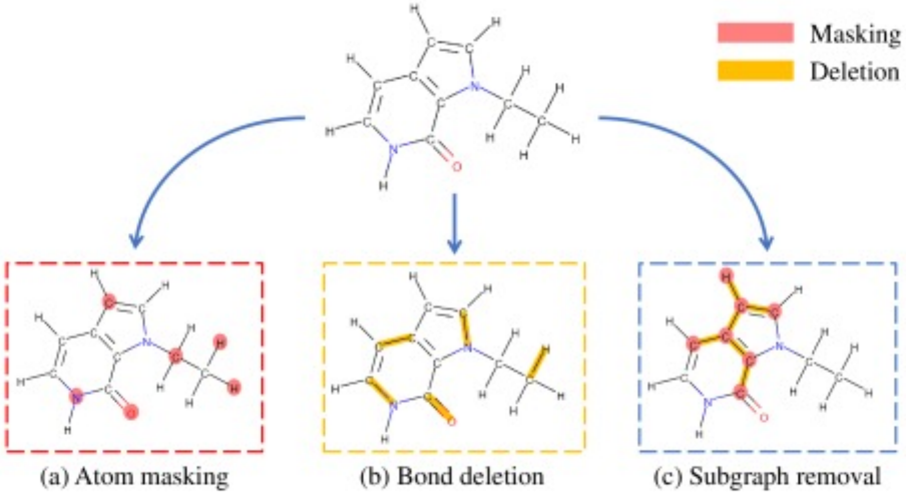
https://www.cambridge.org/core/books/abs/applying-graph-theory-in-ecological-research/spatiotemporal-graphs/ED048111EDD5E344DC897C7D305069A5
https://ai.googleblog.com/2023/03/teaching-old-labels-new-tricks-in.html

Future Directions

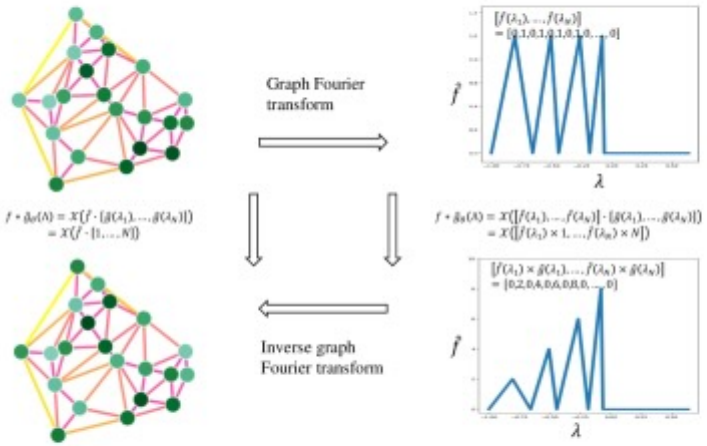
- Augmentation for Graph Contrastive Learning

Existing augmentations: Feature and/or structure perturbing.

Can we develop more effective augmentation strategy for graphs?



Knowledge-based augmentation

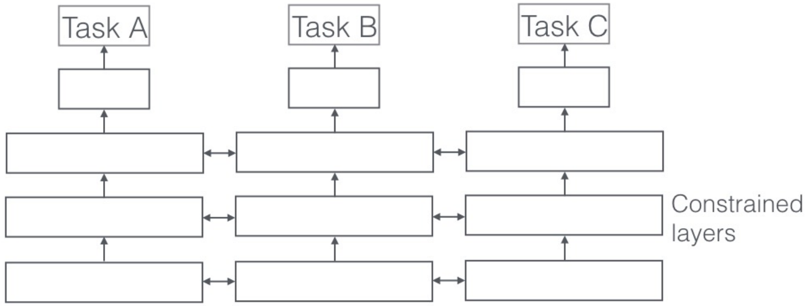


Spectral-based augmentation

Wang, Yuyang, et al. "Molecular contrastive learning of representations via graph neural networks." Nature Machine Intelligence 4.3 (2022): 279-287.
https://deepai.org/publication/analysis-of-irregular-spatial-data-with-machine-learning-classification-of-building-patterns-with-a-graph-convolutional-neural-network

Future Directions

- Learning with Multiple Pretext Tasks



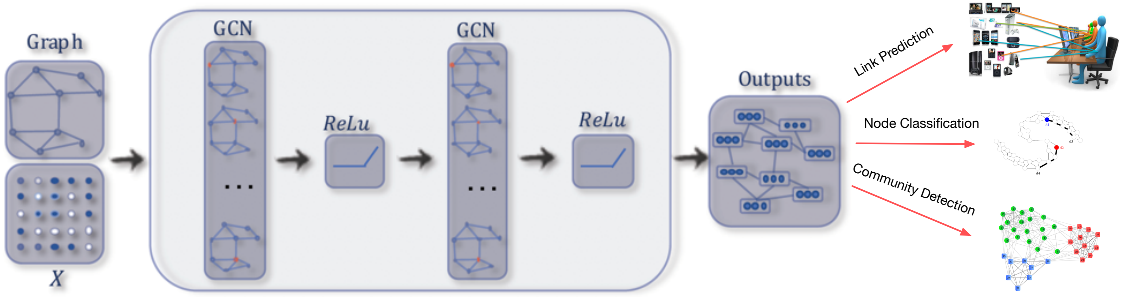
How to effectively leverage different pretext tasks?
Can we select pretext tasks automatically?

- Broader Scope of Applications

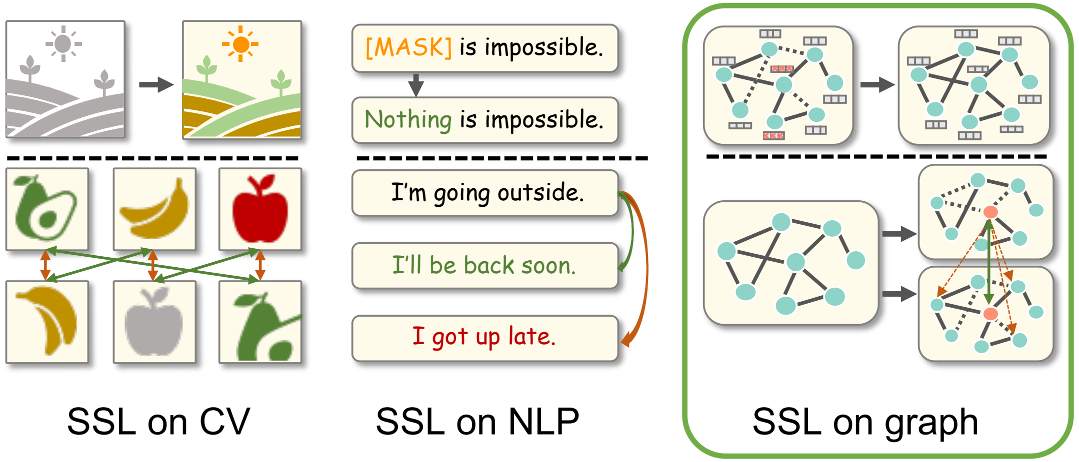
Can we apply GSSL to more graph-related scenarios?

Conclusion

- Background



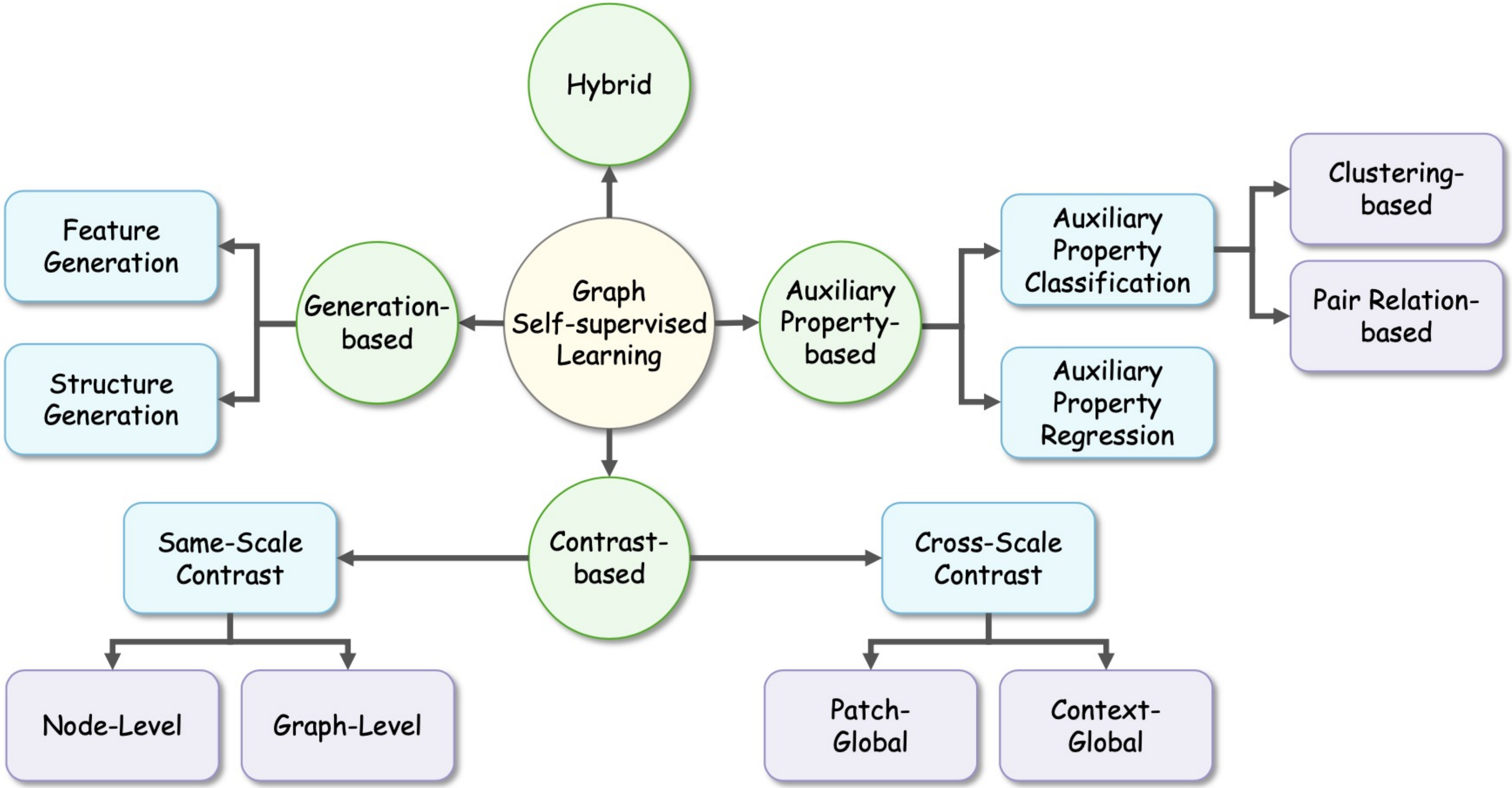
Graph neural networks



Self-supervised learning on graph:
acquires supervision signals from data itself for graph-based deep learning models.

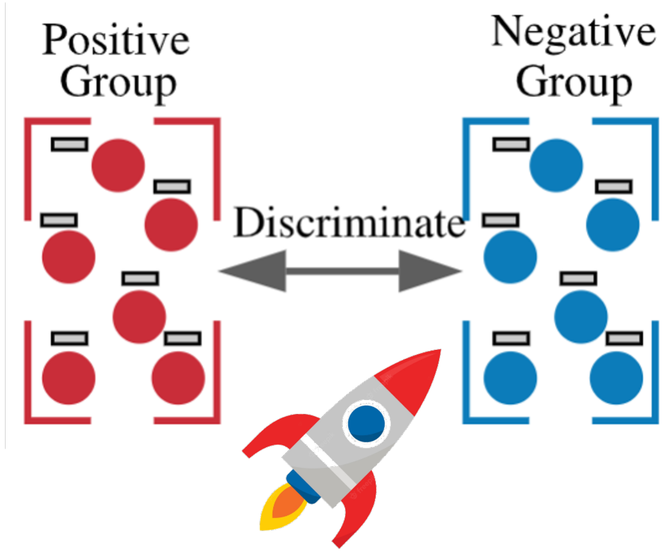
Conclusion

- Graph self-supervised learning: Taxonomy

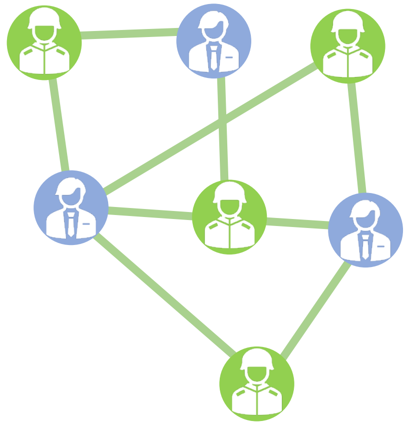


Conclusion

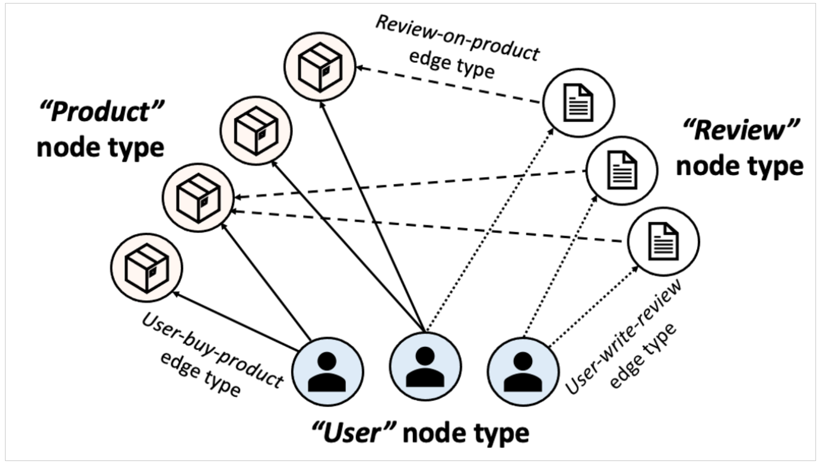
- Graph self-supervised learning: Frontiers



Efficient GSSL paradigm:
Group Discrimination



GSSL for
Heterophilic graph



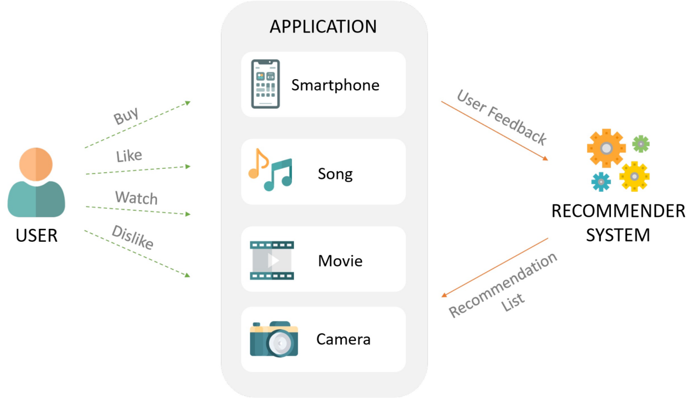
GSSL for
Heterogeneous graph

https://ai.googleblog.com/2023/03/teaching-old-labels-new-tricks-in.html

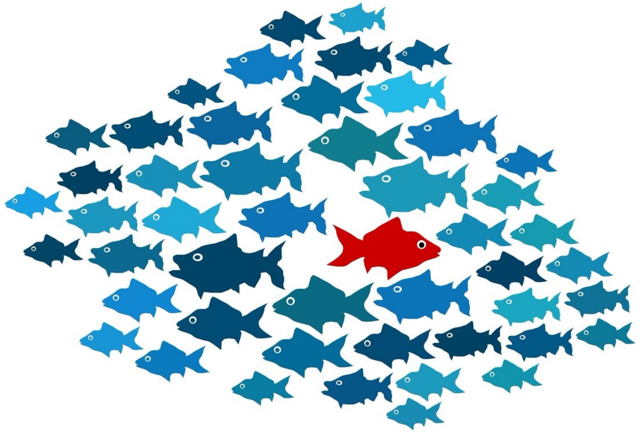
Conclusion

- Graph self-supervised learning: Applications

- **Recommender Systems**

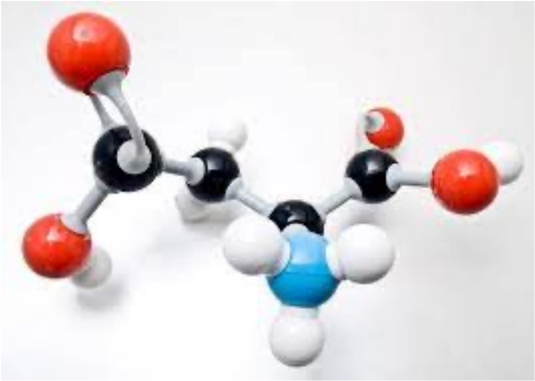


- **Outlier Detection**

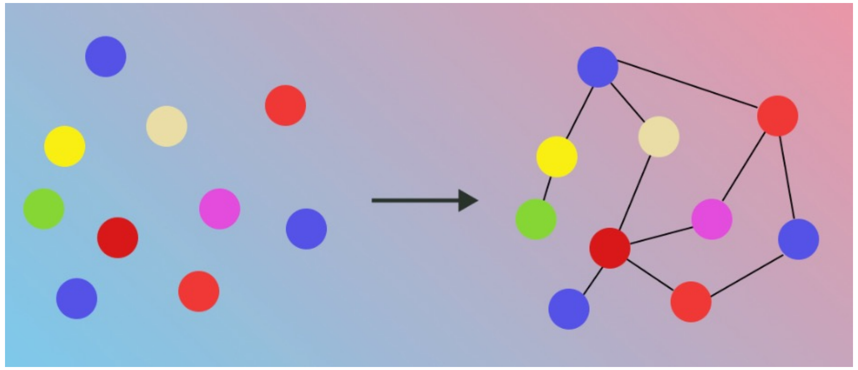


Boarder Applications…

- **Chemistry**



- **Graph Structure Learning**



Thanks for listening!
Q&A