Towards Graph Foundation Models

WWW 2024 Tutorial

Philip S. Yu, Chuan Shi, Cheng Yang, Yuan Fang, Lichao Sun

北京郵電大學
Beijing University of Posts and Telecommunications

ck
L L ck ck

Cheng Yang

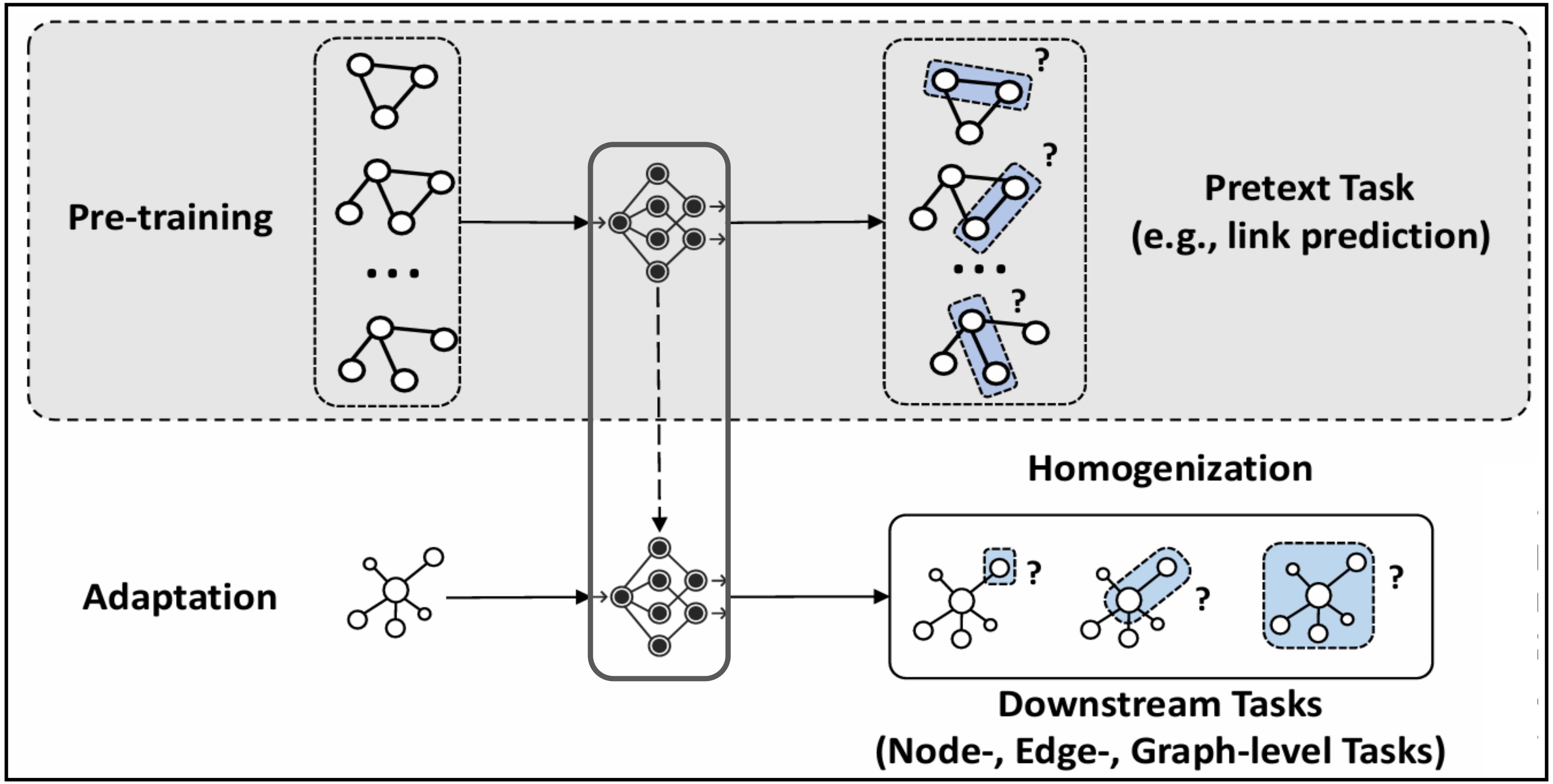
yangcheng@bupt.edu.cn

Beijing University of Posts and Telecommunications

: L ck ck ck

: ck

(ck ck ck)

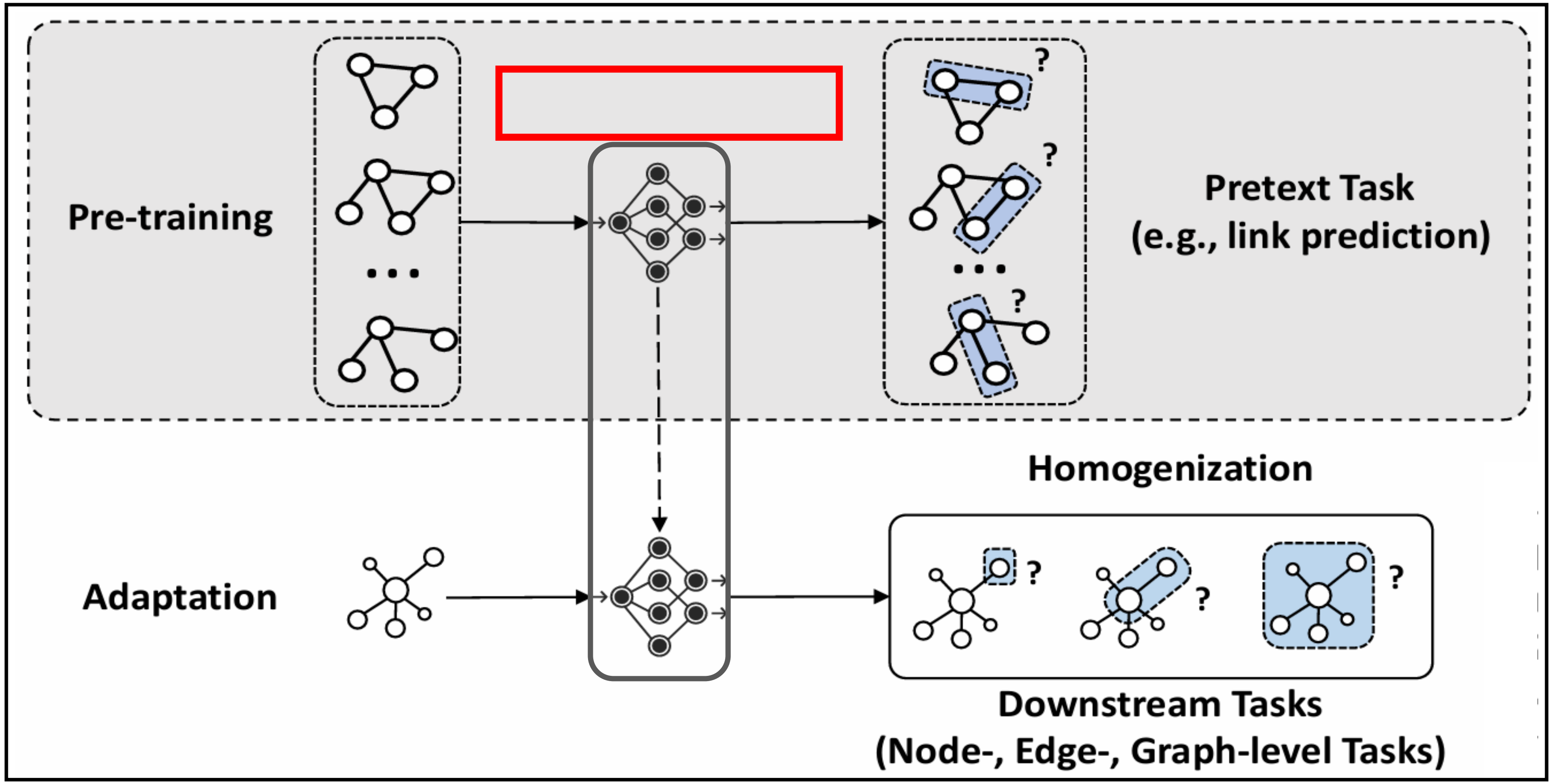


Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

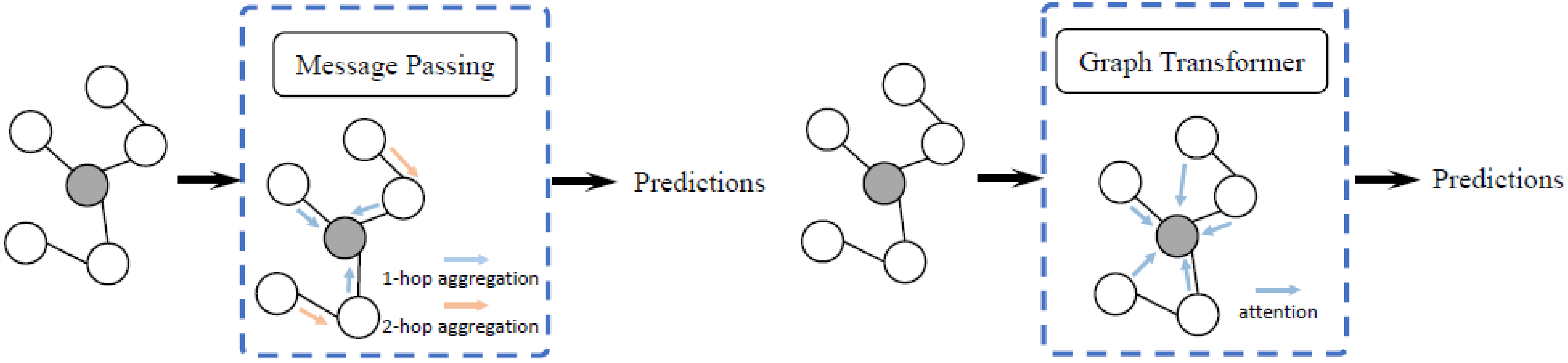
: L ďk ďk ďk

: ďk

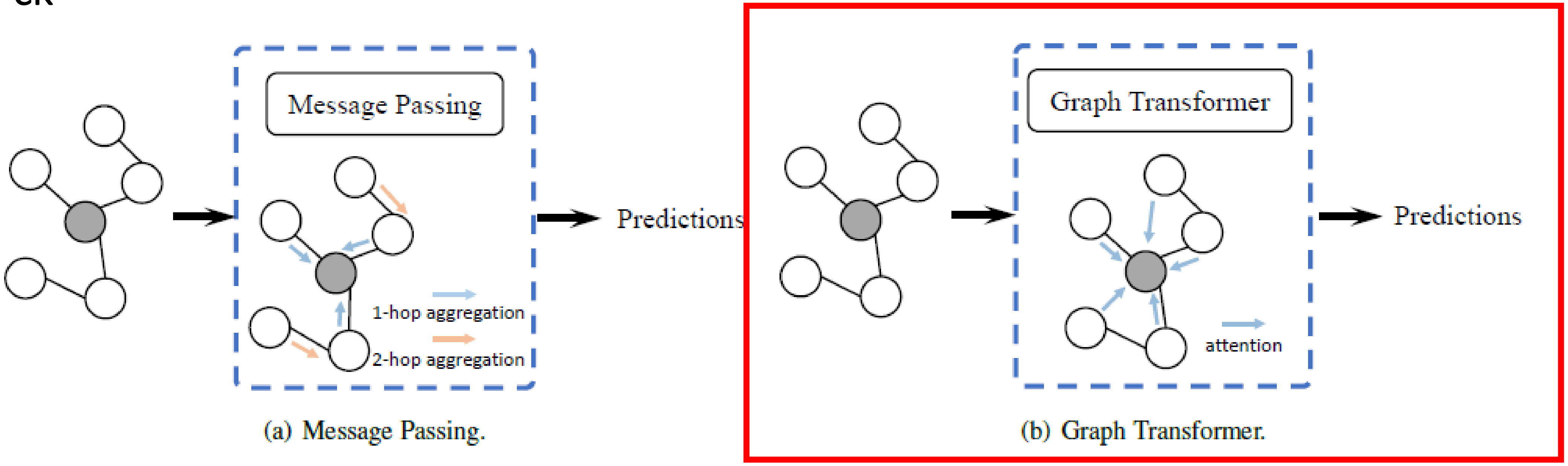
(ďk ďk ďk)



Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023



(a) Message Passing. (b) Graph Transformer.



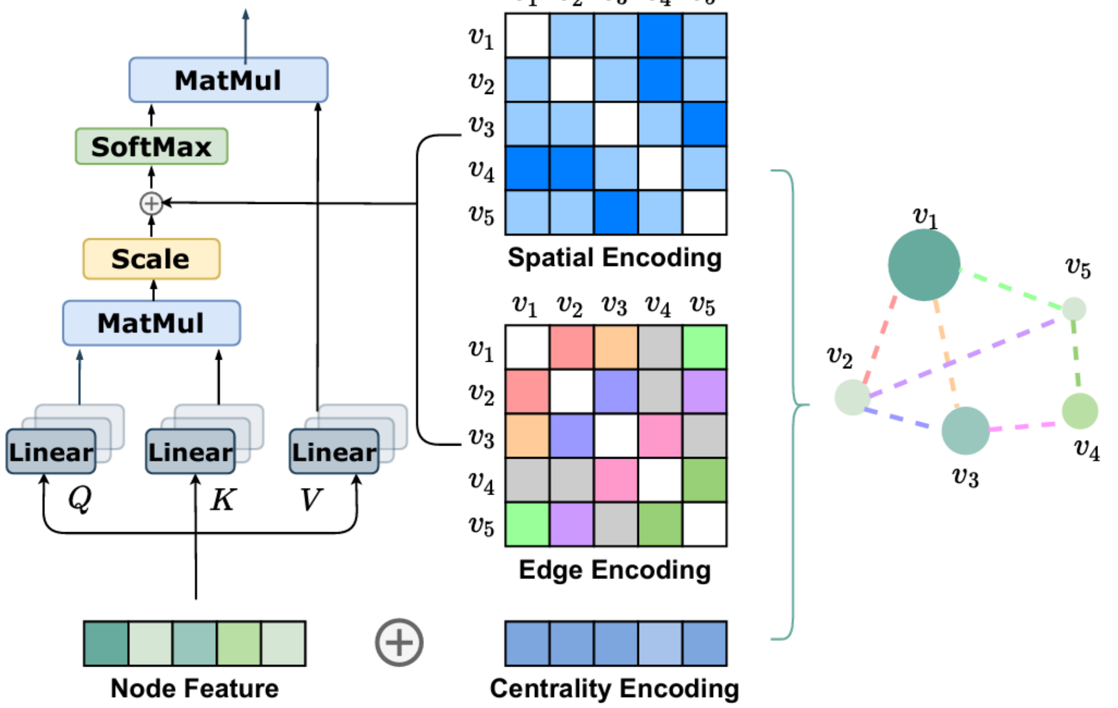
(a) Message Passing.

(b) Graph Transformer.

Whether Transformer architecture is suitable to model graphs and how to make it work in graph representation learning?

ck ck :

- ck ck
  ck ck
  - ck ck ck
    ck ck ck ck
  - ck ckP ck
    ck ck ck ckck ck
  - ck ck ck ck ck
    ck ck ck



P :

-
    -

$$h_i^{(0)} = x_i + z^{-}_{\deg^{-}(v_i)} + z^{+}_{\deg^{+}(v_i)}$$

- P
    - P B
-
    - SP_{ij}

$$A_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d}} + b_{\phi(v_i,v_j)} + c_{ij}, \text{ where } c_{ij} = \frac{1}{N}\sum_{n=1}^{N} x_{e_n}(w_n^E)^T$$

:

- ďk ďk ďk ďk ďk ďk ďk ďkďk ďk
- ďk ďkďk ďk ďk ďk ďk ďk ďk ďk ďk ďk
  ďk ďk ďk
- L L ďk ďk ďk ďk ďk ďk ďk ďk ďk ďk
  ďk ďk ďk ďk

ck ck ck ckck

L ck ck ck

ck ck ck ck ck

ck ckck

ck ckck ck ck ckck ck

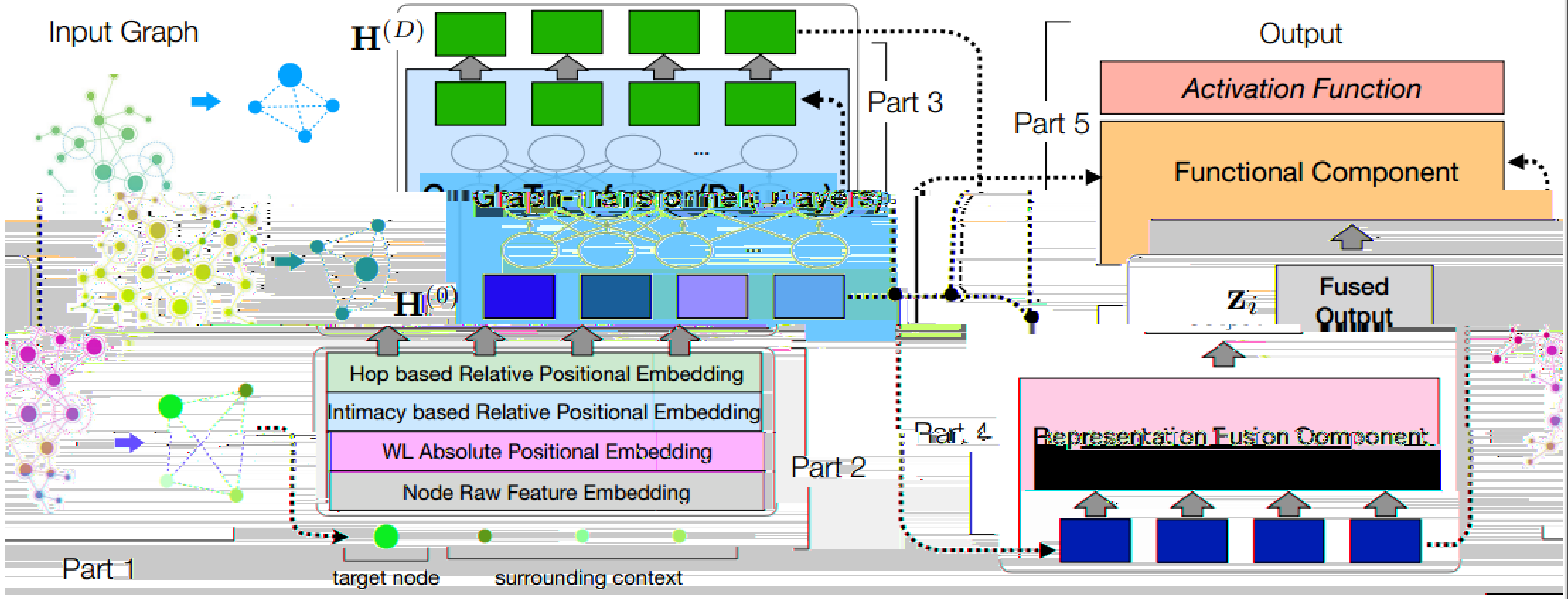
ck

ck ck ck ck ck

ck ck ck ckck ck

ck ck

ck ck



$$\mathbf{e}_j^{(x)} = \text{Embed}\left(\mathbf{x}_j\right) \in \mathbb{R}^{d_h \times 1}$$

$$\mathbf{e}_j^{(r)} = \text{Position-Embed}\left(\text{WL}(v_j)\right)$$

$$= \left[sin\left(\frac{\text{WL}(v_j)}{10000^{\frac{2l}{d_h}}}\right), cos\left(\frac{\text{WL}(v_j)}{10000^{\frac{2l+1}{d_h}}}\right)\right]_{l=0}^{\left\lfloor \frac{d_h}{2} \right\rfloor}$$

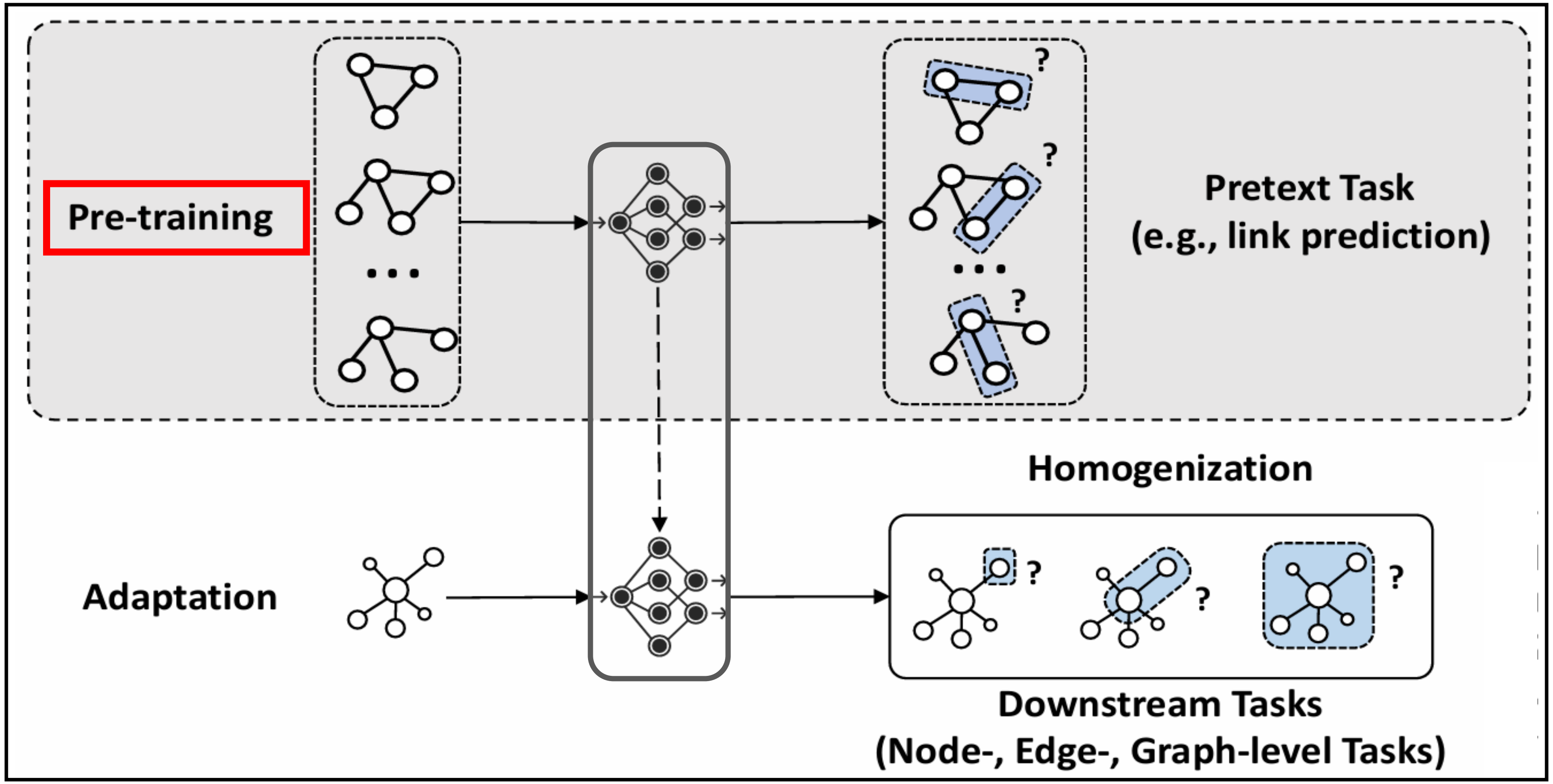
$$\mathbf{e}_j^{(p)} = \text{Position-Embed}\left(\text{P}(v_j)\right) \in \mathbb{R}^{d_h \times 1}$$

$$\mathbf{e}_j^{(d)} = \text{Position-Embed}\left(\text{H}(v_j; v_i)\right) \in \mathbb{R}^{d_h \times 1}.$$

$$\mathbf{h}_j^{(0)} = \text{Aggregate}(\mathbf{e}_j^{(x)}, \mathbf{e}_j^{(r)}, \mathbf{e}_j^{(p)}, \mathbf{e}_j^{(d)})$$

: L ɗk ɗk ɗk : ɗk

(ɗk ɗk ɗk)

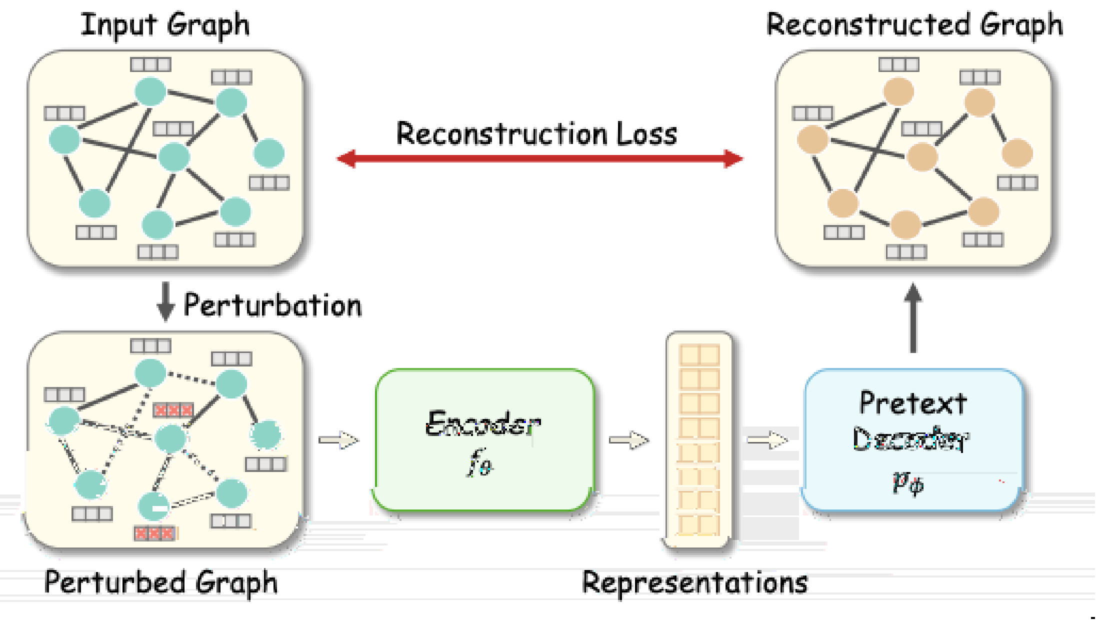


Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

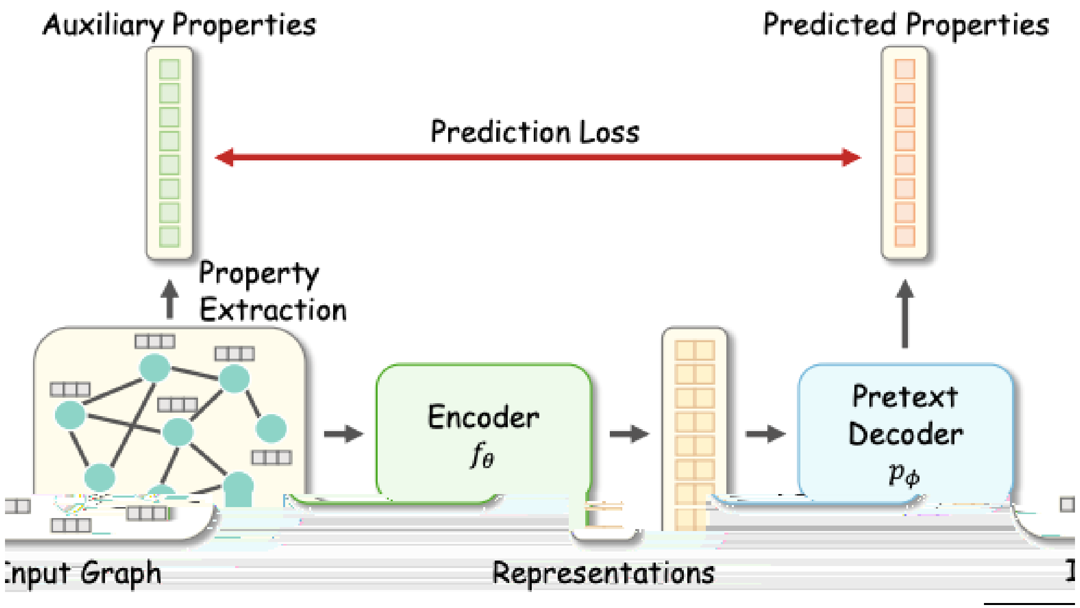
Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

ꝁk

- ꝁk ꝁk ꝁk ꝁk ꝁk ꝁk ꝁk

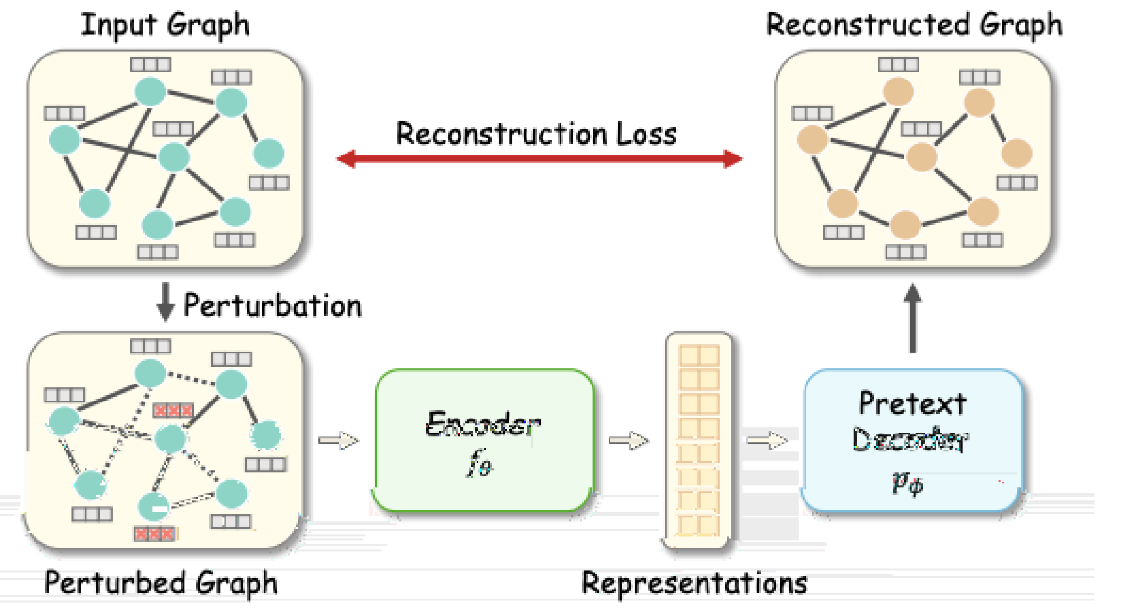


graph reconstruction

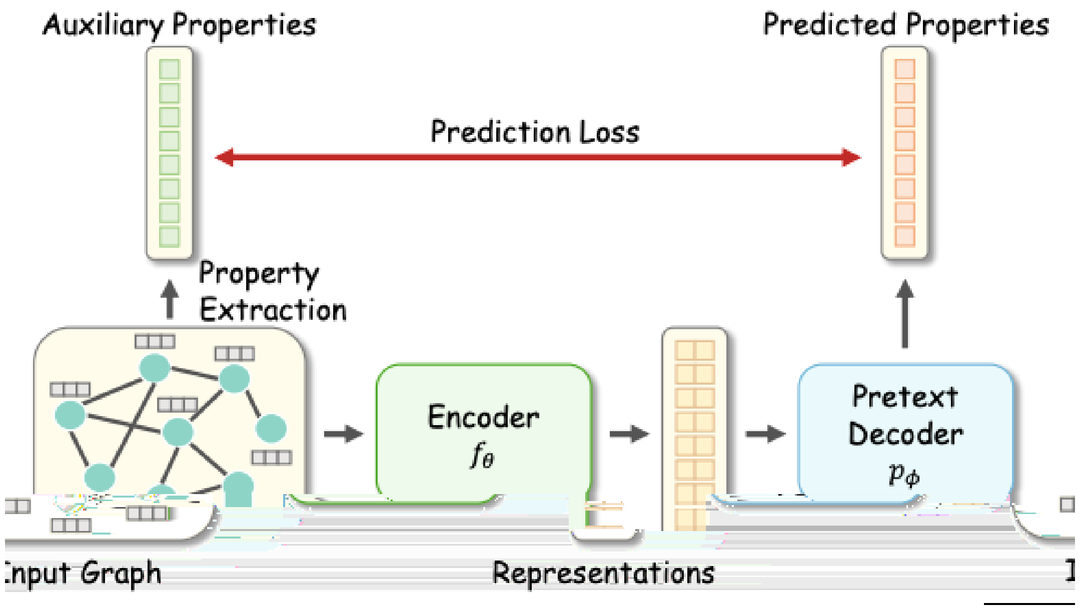


property prediction

Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.



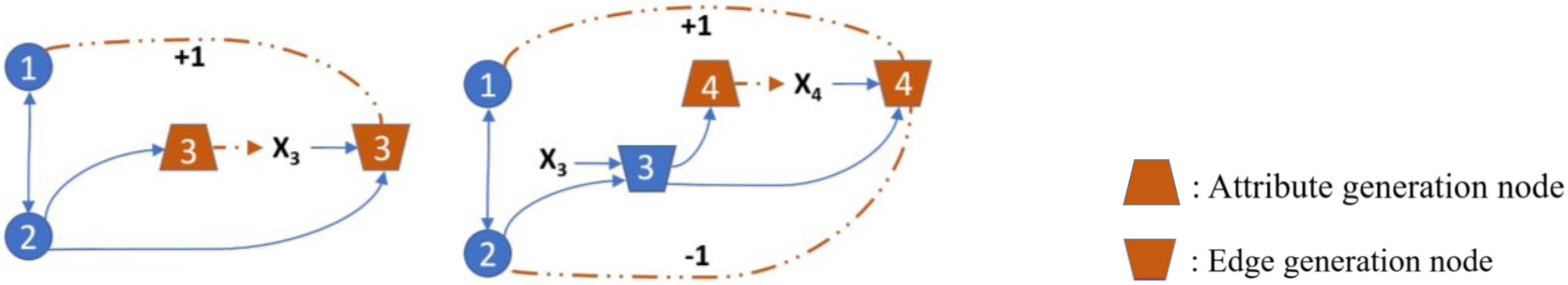
graph reconstruction



property prediction

Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

Hu Z, Dong Y, Wang K, et al. GPT-GNN: Generative pre-training of graph neural networks. KDD2020

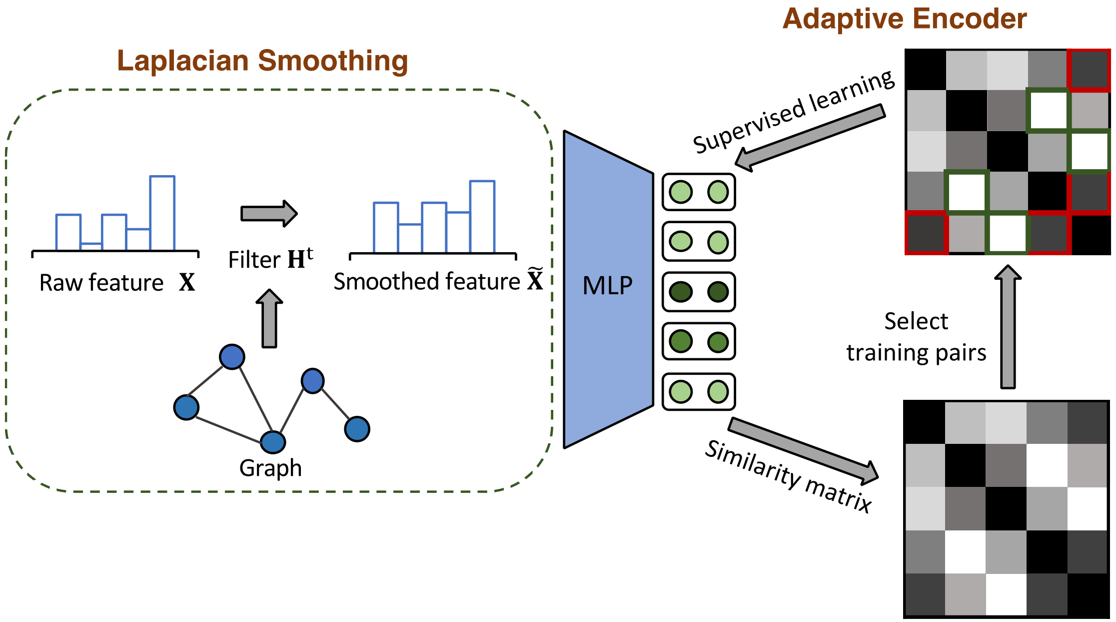


(d) Generate attribute and masked edges for node 3.

(e) Generate attribute and masked edges for node 4.

Hu Z, Dong Y, Wang K, et al. GPT-GNN: Generative pre-training of graph neural networks. KDD2020

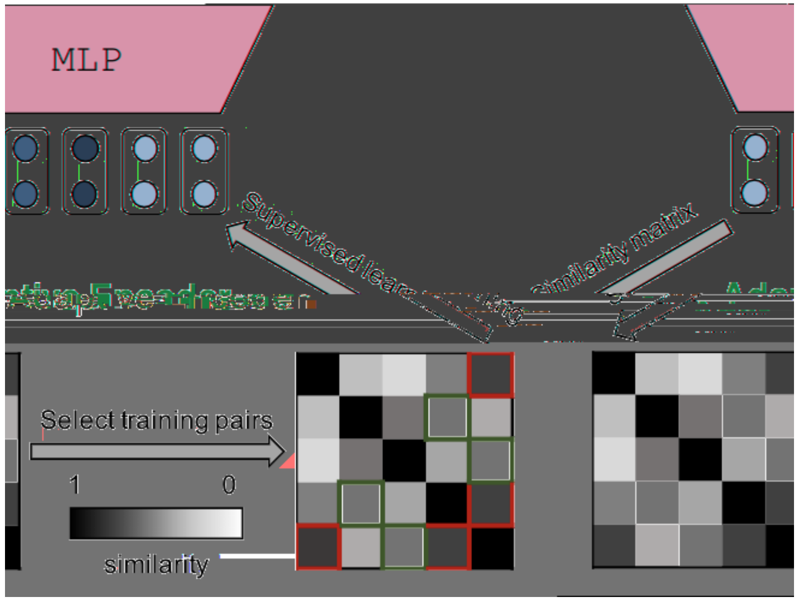
Cui G, Zhou J, Yang C, et al. Adaptive graph encoder for attributed graph embedding. KDD 2020



Adaptive Encoder
Laplacian Smoothing
Supervised learning
Raw feature X
Filter $\mathbf{H}^t$
Smoothed feature $\tilde{\mathbf{X}}$
MLP
Graph
Select
training pairs
Similarity matrix

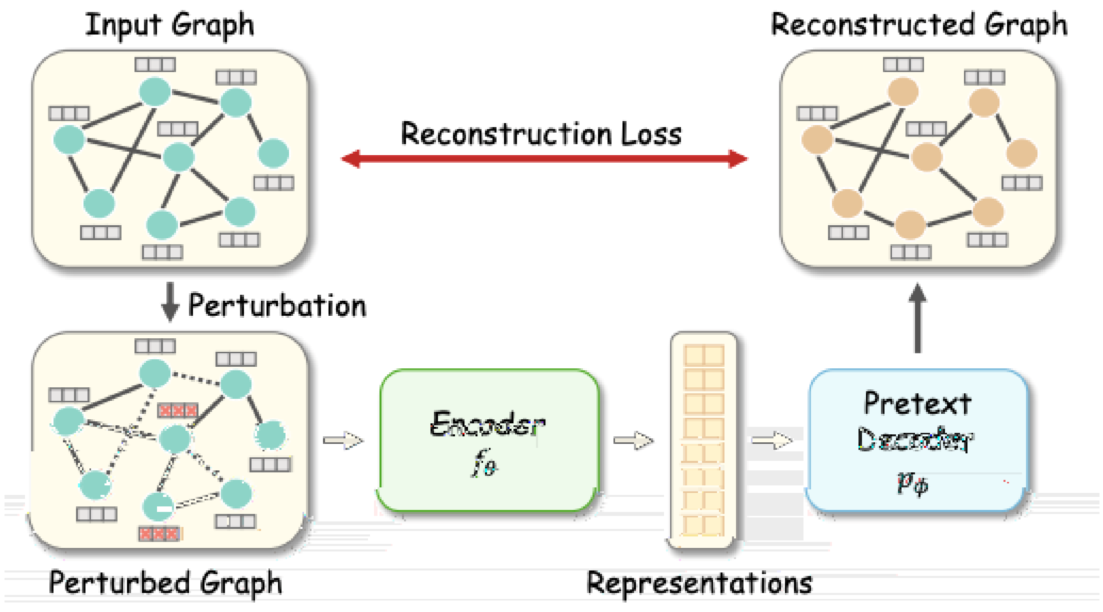
- P

- P
- B

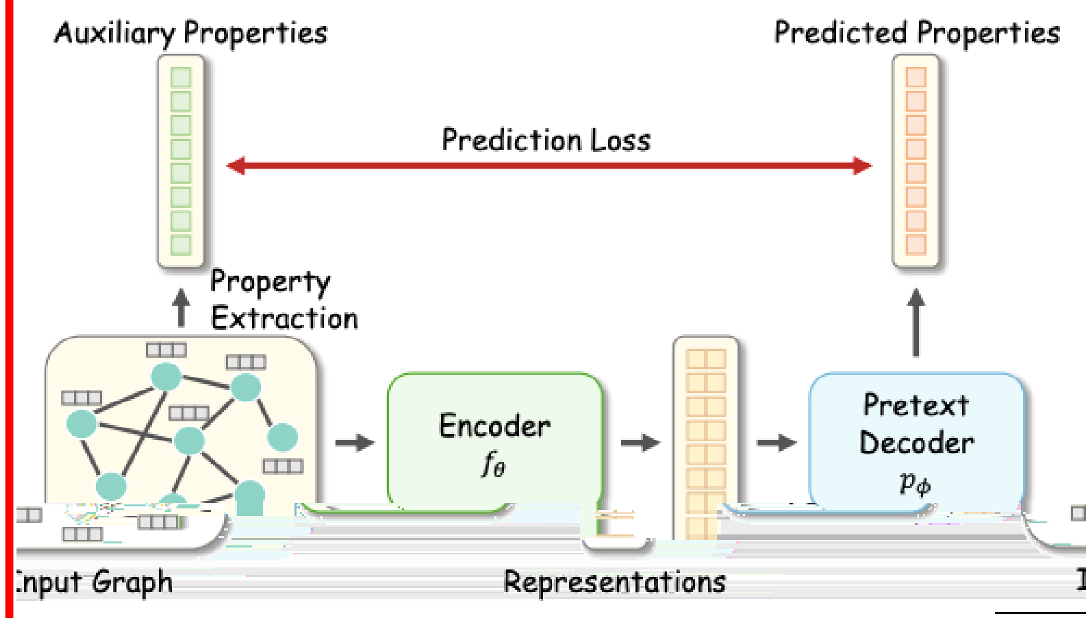


Cui G, Zhou J, Yang C, et al. Adaptive graph encoder for attributed graph embedding. KDD 2020 ꝺk

- ꝺk ꝺk ꝺk ꝺk ꝺk ꝺk ꝺk



graph reconstruction

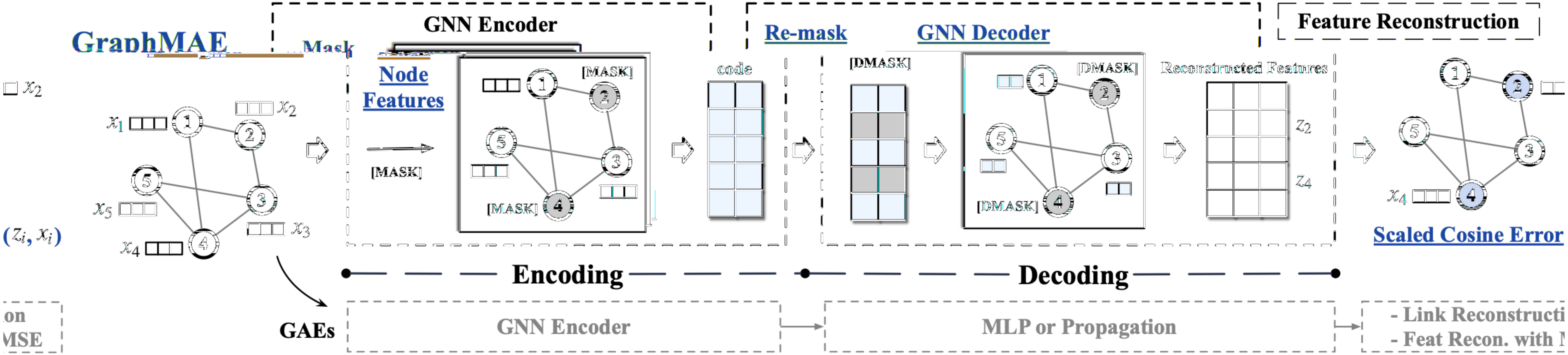


property prediction

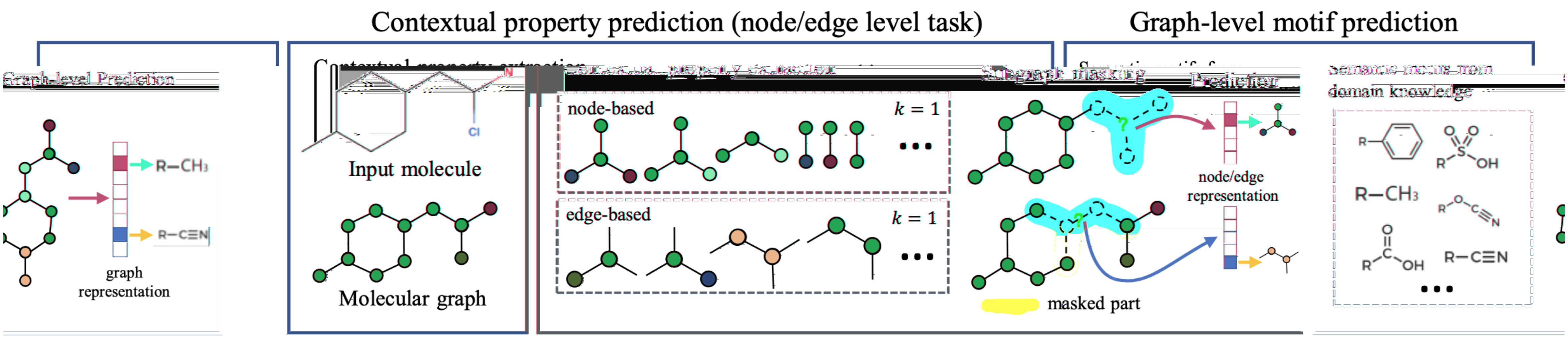
Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

Hou Z, Liu X, Cen Y, et al. Graphmae: Self-supervised masked graph autoencoders[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.

ck ck ck ck ck ck
ck ck ck ck ck ck
ck ck ckck ck ck ck ck ck
ck ck
ck ck ck ckck ck ck ck
ck ck L L ck ck ck ck
ck ck ck ck

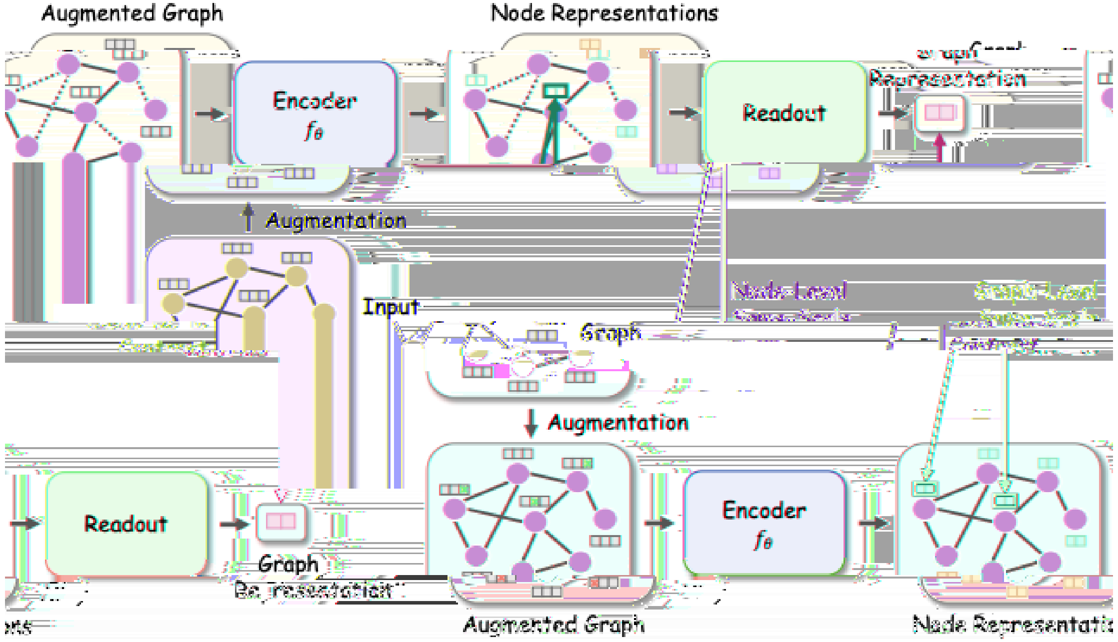


Yu Rong, Yatao Bian, et al. Self-Supervised Graph Transformer on Large-Scale Molecular Data. NIPS2020.

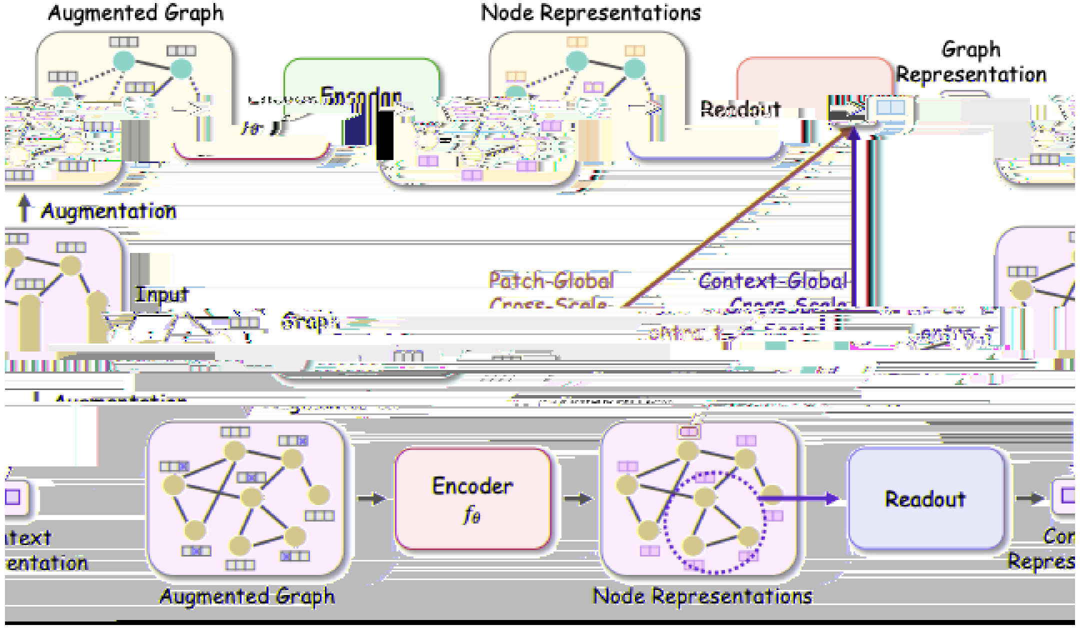

Contextual property prediction (node/edge level task)
Graph-level motif prediction
Input molecule
Molecular graph
node-based
k = 1
edge-based
k = 1
node/edge representation
masked part
graph representation
R—CH3
R—C≡N
Cl
N
domain knowledge

ck

- ck ck ck ck ckck ck ckck



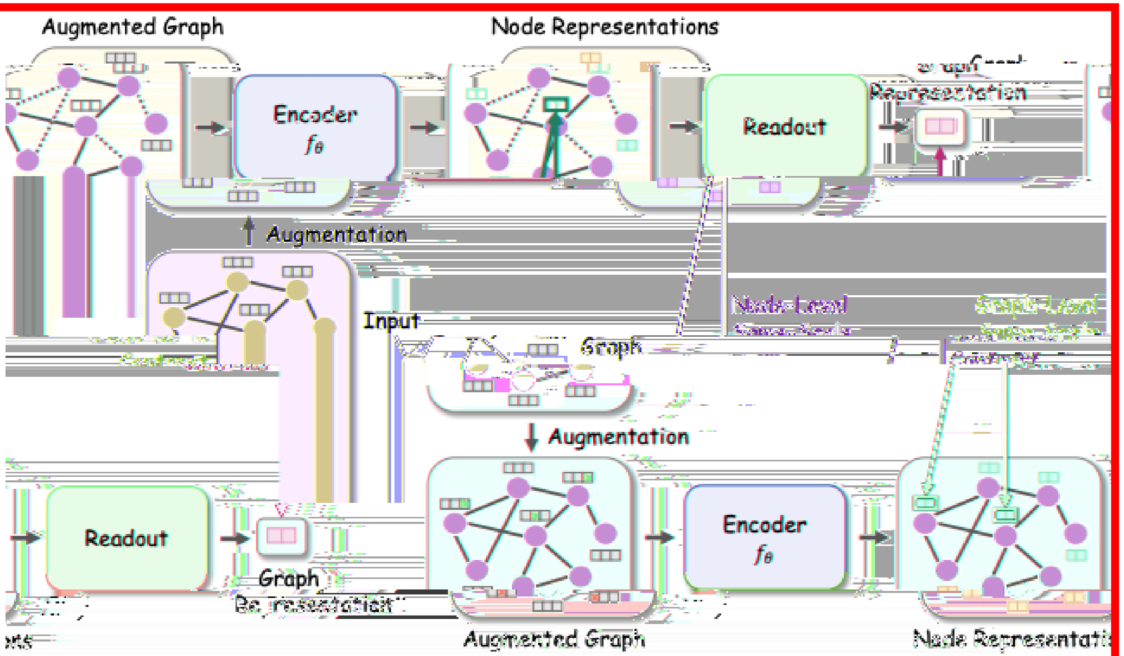
same-scale contrastive learning



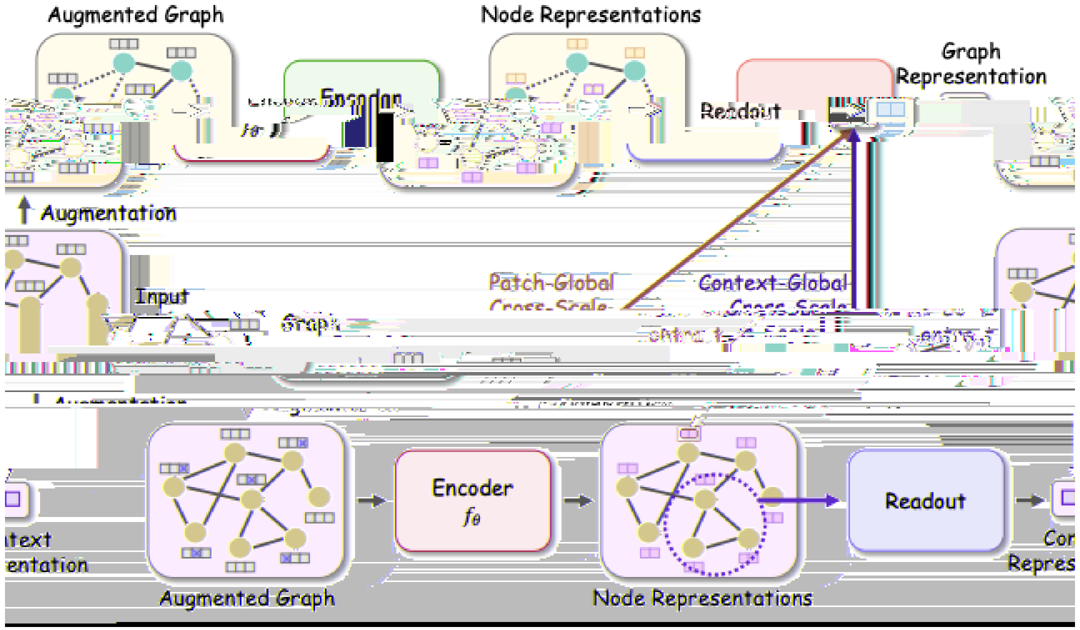
cross-scale contrastive learning

Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

•



same-scale contrastive learning



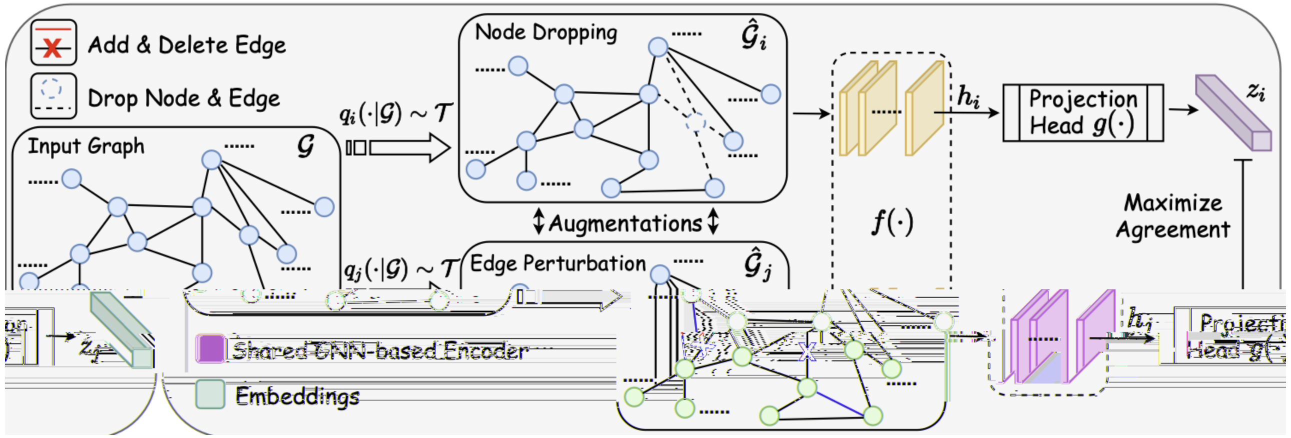
cross-scale contrastive learning

Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

- ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk L L ƈk L L
- ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk
- ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈk ƈkƈk ƈk

B ck ck ck ckck

ck ck ck ck ck ck ck ck ck



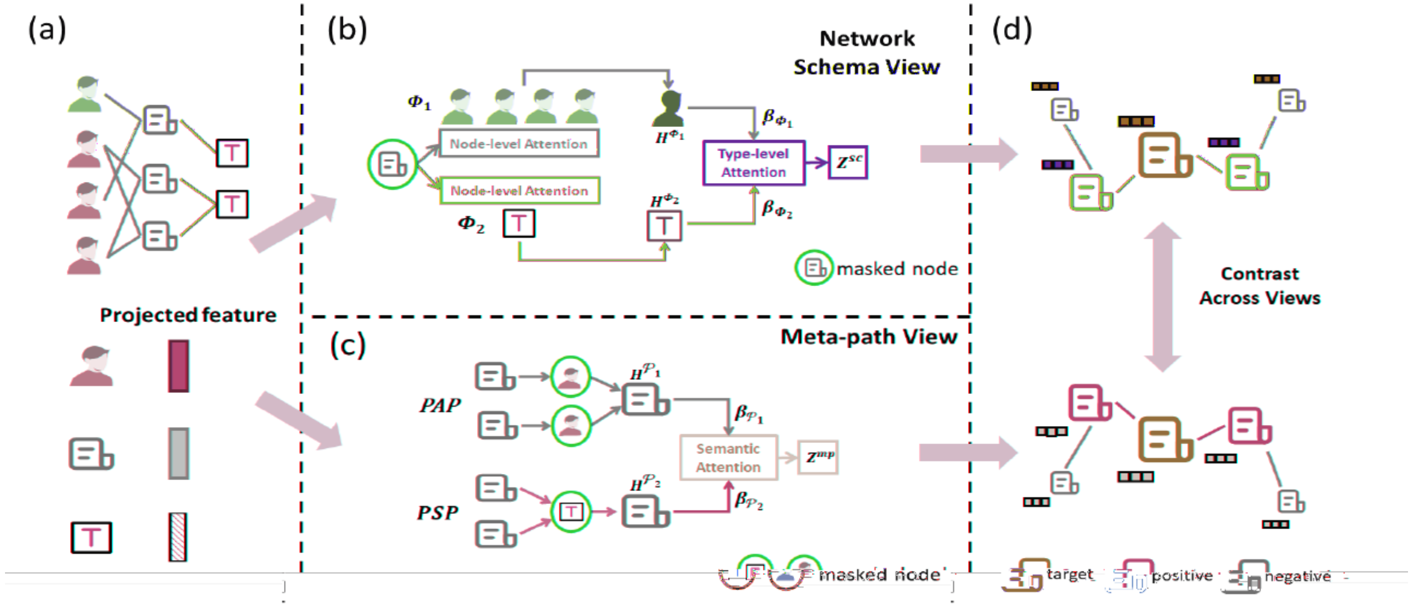
$$\ell_n = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n,j})/\tau)}{\sum_{n'=1, n' \neq n}^{N} \exp(\mathrm{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n',j})/\tau)}$$

ck ck ckck

Data augmentation	**Type**	**Underlying Prior**
Node dropping	Nodes, edges	Vertex missing does not alter semantics.
Edge perturbation	Edges	Semantic robustness against connectivity variations.
Attribute masking	Nodes	Semantic robustness against losing partial attributes.
Subgraph	Nodes, edges	Local structure can hint the full semantics.

ďk ďk

- ďk L ďk ďk ďk ďk ďk ďk ďk ďk ďk ďk
- ďk ďk ďk ďk ďk ďk ďk ďk ďk ďkďk ďk ďk
- ďk ďk ďk ďk ďkďk ďk ďk ďk ďk ďkďk ďk



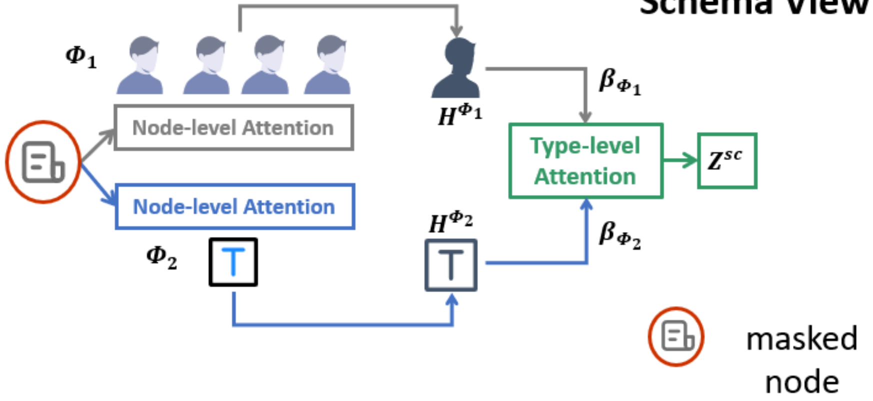
L ďk Pďk ďk ďk ďkďk ďk ďk ďk ďk ďk ďk B B ɗk ɗk

ɗkΦ_m ɗk ɗk ɗk T_{Φ_m}

ɗk ɗk ɗk ɗk ɗk ɗk ɗk ɗkΦ_m

ɗk ɗk ɗkɗk ɗk ɗk ɗk ɗk

Network Schema View



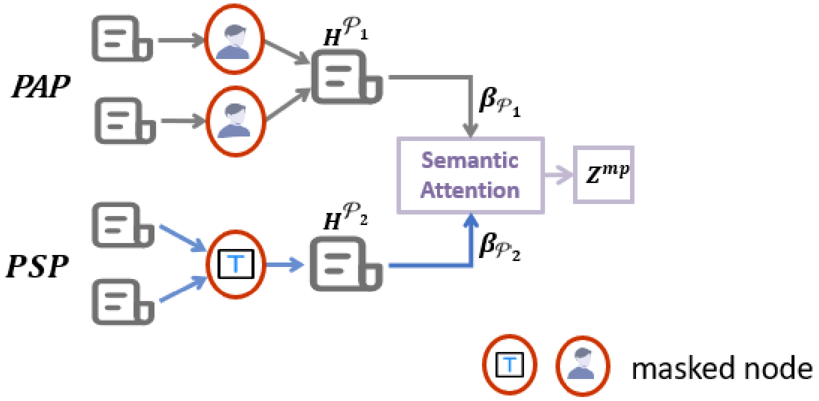
ɗk ɗk

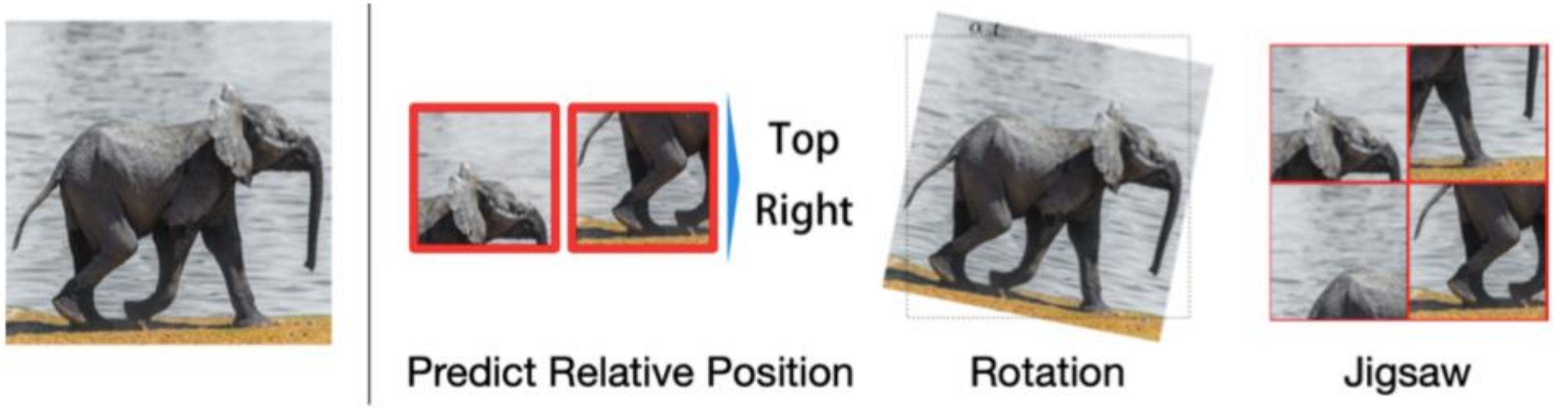
ɗk ɗk ɗkɗk ɗk

ɗk ɗk ɗk ɗk ɗk L

ɗk ɗkɗk ɗk ɗk ɗkɗk ɗk
ɗk ɗk

Meta-path View





Can we generate better augmentations than typical random dropping-based methods?

ck ck

- ck ck ck L L ck ck ck ck *g* ck *h*
  - ck *g* ck ck ck
  - ck *h* ck ck

$$g(\boldsymbol{Z};\boldsymbol{F}) = \boldsymbol{F}\boldsymbol{Z},\ h(\boldsymbol{Z};\boldsymbol{W}) = \sigma(\boldsymbol{Z}\boldsymbol{W}), \qquad \boldsymbol{F} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$$

$$GCN(\boldsymbol{X}) = h_L \circ g \circ h_{L-1} \circ g \circ \cdots \circ h_1 \circ g(\boldsymbol{X}),$$

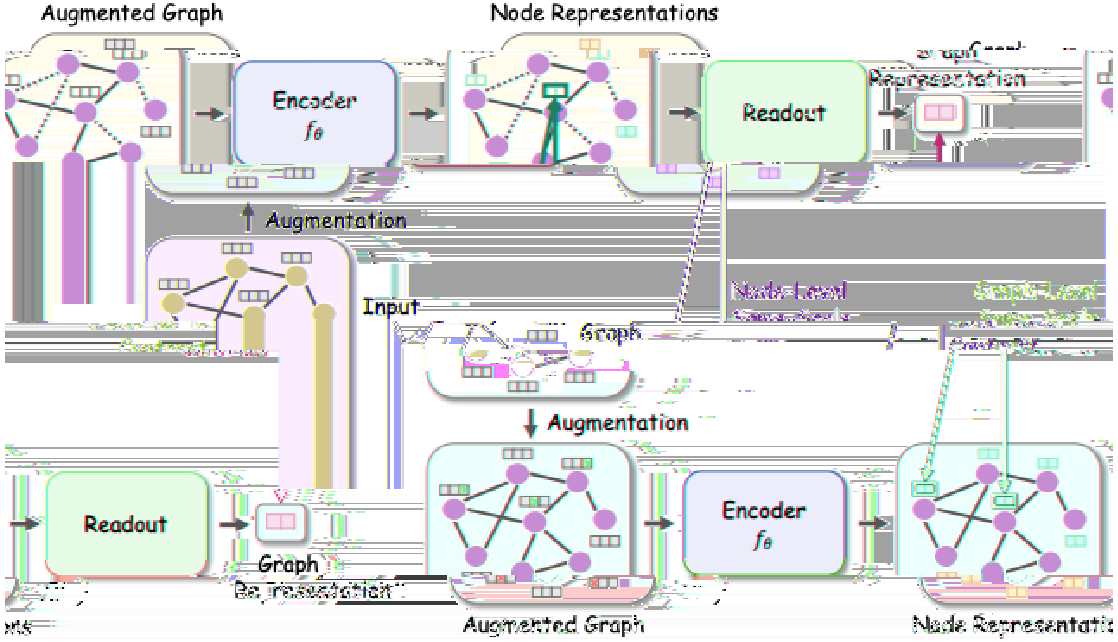
$$SGC(\boldsymbol{X}) = h \circ g^{[L]}(\boldsymbol{X}),$$

- ckck ck ck ck ck ck ck ck ck ck ck ck
- ck ck ck ck ck ck ck ck ck ck

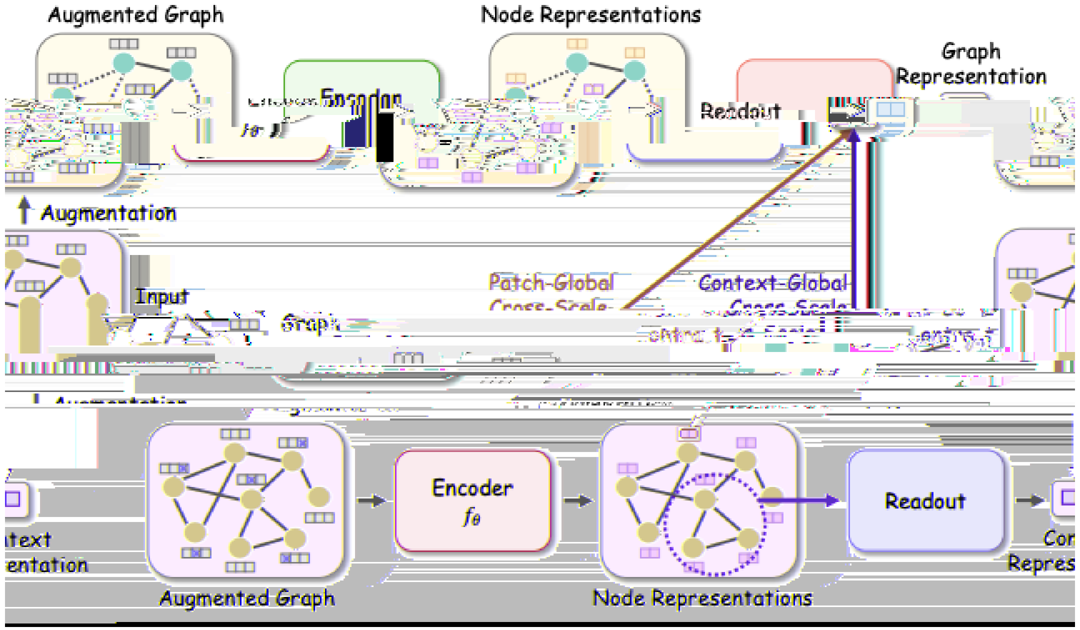
ƈk ƈk ƈkƈk ƈk ƈk ƈk ƈk

- ƈk ƈk
  - ƈk ƈk ƈk ƈk ƈk *h* ƈk ƈkƈk ƈkƈk ƈk
  - ƈk ƈkƈk ƈk ƈk *g* ƈkƈk ƈk
- ƈk
  - ƈk ƈk ƈk *g* ƈk ƈk ƈk
- P ƈk
  - ƈk ƈk ƈk ƈk

Datasets	Cora	CiteSeer	PubMed	Coauthor-CS	Amazon-C	Amazon-P	Avg. Acc.	Avg. Rank
GCN	82.5 ± 0.4	71.2 ± 0.3	79.2 ± 0.3	93.03 ± 0.3	86.51 ± 0.5	92.42 ± 0.2		
GAT	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	92.31 ± 0.2	86.93 ± 0.3	92.56 ± 0.4	-	-
InfoGCL	83.5 ± 0.3	73.5 ± 0.4	79.1 ± 0.2	-	-	-		
DGI	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.3	92.15 ± 0.6	83.95 ± 0.5	91.61 ± 0.2	83.10	8.5
GRACE	81.7 ± 0.4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	84.44	6.5
MVGRL	83.4 ± 0.3	73.0 ± 0.3	80.1 ± 0.6	92.11 ± 0.1	87.53 ± 0.1	91.74 ± 0.0	[illegible]	[illegible]
BGRL	81.7 ± 0.5	72.1 ± 0.5	80.2 ± 0.4	93.01 ± 0.2	88.23 ± 0.3	92.57 ± 0.3	84.63	6.5
GCA	83.4 ± 0.3	72.3 ± 0.1	80.2 ± 0.4	93.10 ± 0.0	87.85 ± 0.3	92.53 ± 0.2	84.89	4.0
SimGRACE	77.3 ± 0.1	71.4 ± 0.1	78.3 ± 0.3	93.45 ± 0.4	86.04 ± 0.2	91.39 ± 0.4	82.98	8.5
COLES	81.2 ± 0.4	71.5 ± 0.2	80.4 ± 0.7	92.65 ± 0.1	79.64 ± 0.0	89.00 ± 0.5	82.49	8.8
ARIEL	82.5 ± 0.1	72.2 ± 0.2	80.5 ± 0.3	93.35 ± 0.0	88.27 ± 0.2	91.43 ± 0.2	84.71	4.8
CCA-SSG	83.9 ± 0.4	73.1 ± 0.3	81.3 ± 0.4	93.37 ± 0.2	88.42 ± 0.3	92.44 ± 0.1	85.42	2.3
Base Model	81.1 ± 0.4	71.4 ± 0.1	79.1 ± 0.4	92.86 ± 0.3	87.65 ± 0.2	91.19 ± 0.3	83.88	9.0
MA-GCL	83.3 ± 0.4	**73.6 ± 0.1**	**83.5 ± 0.4**	**94.19 ± 0.1**	**88.83 ± 0.3**	**93.80 ± 0.1**	**86.20**	**1.2**



same-scale contrastive learning

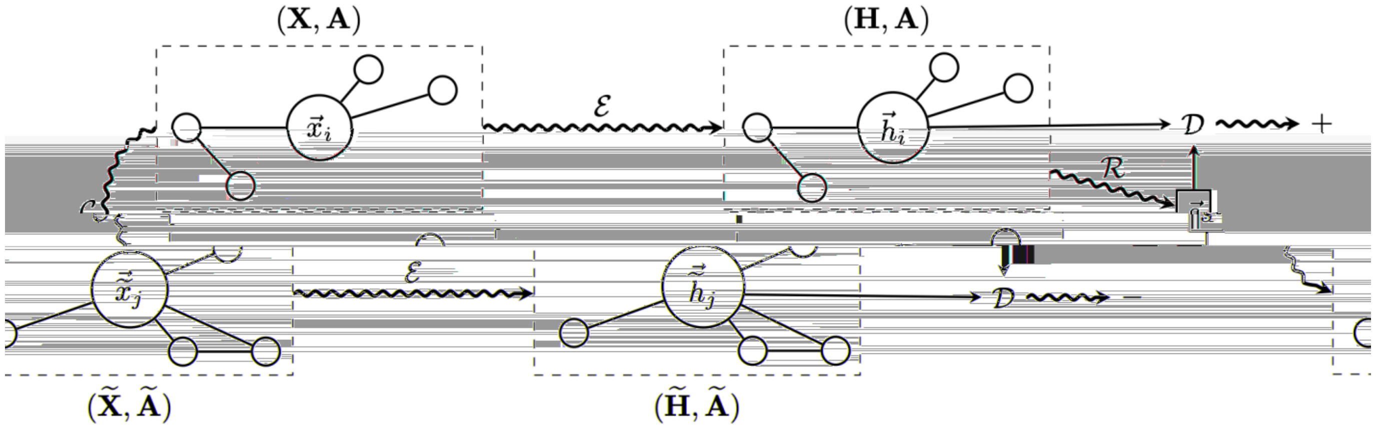


cross-scale contrastive learning

Liu Y, Jin M, Pan S, et al. Graph self-supervised learning: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6): 5879-5900.

Veličković P, Fedus W, Hamilton W L, et al. Deep Graph Infomax[C]//ICLR. 2018.

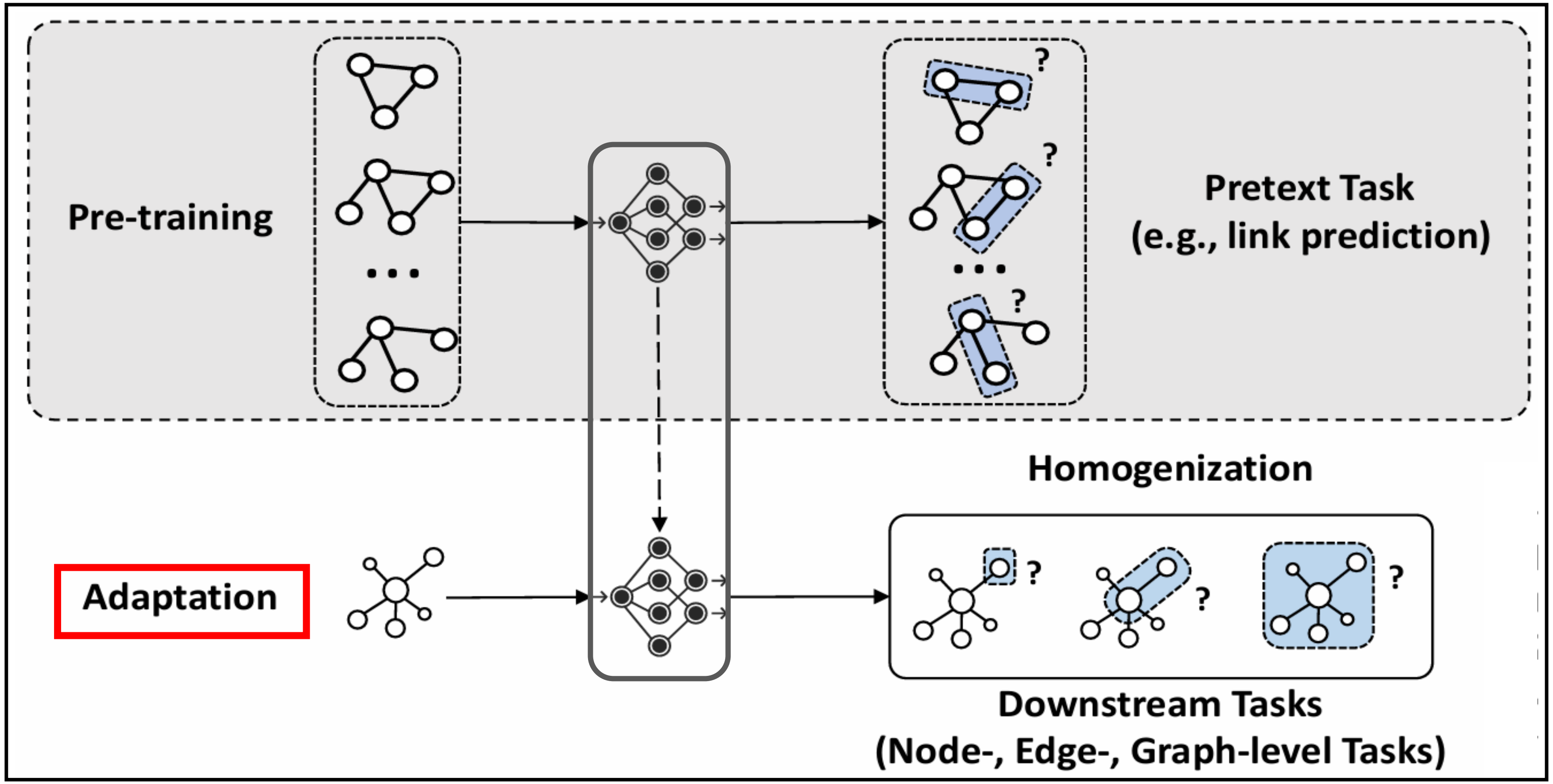
- L ck ck ck ck ck ck ck ck ck
- ck ck ck ck ck ck
  ck ck ck ck ck ck
- ck ck ck ck ck ck ck ck ck ck ck
  ck ck ck ck ck ck ck ck ck ck
- ck ck ck ck
  ck ck ck ck ck ck ck ck ck ck ck ck
  ck ck ck ck ck ck ck



: L ck ck ck

: ck

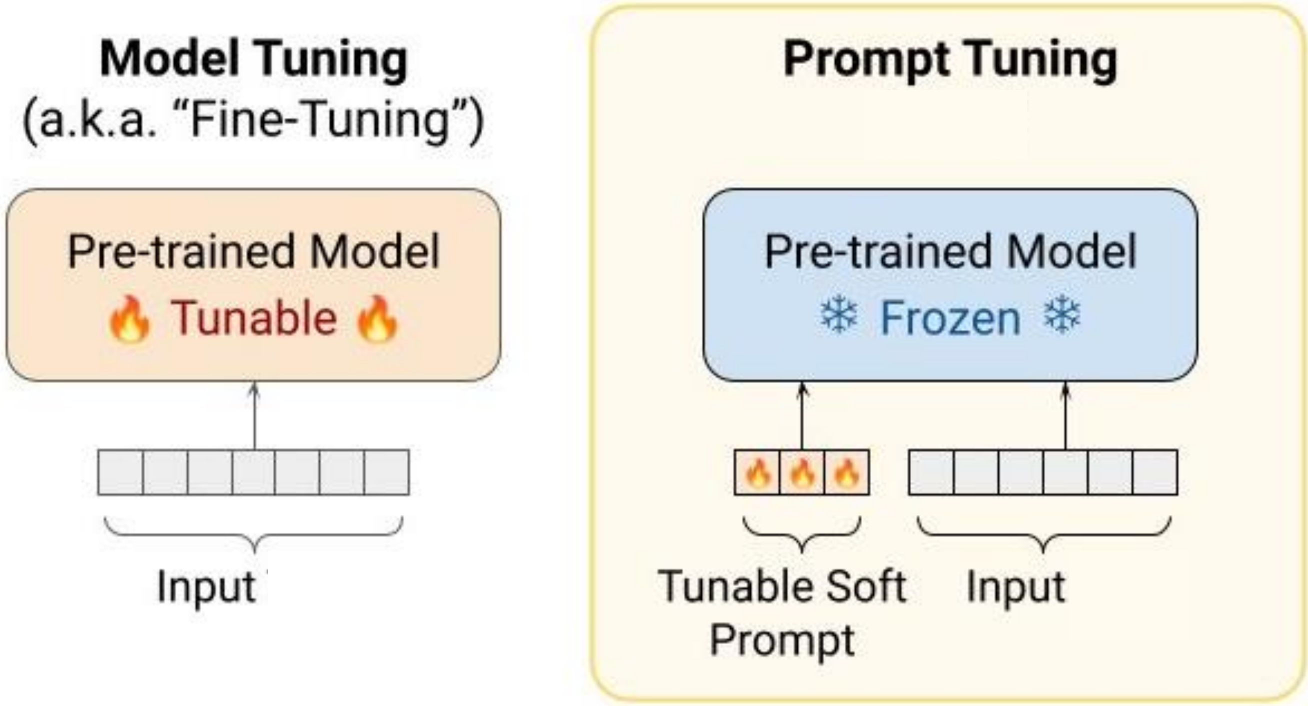
(ck ck ck)



Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

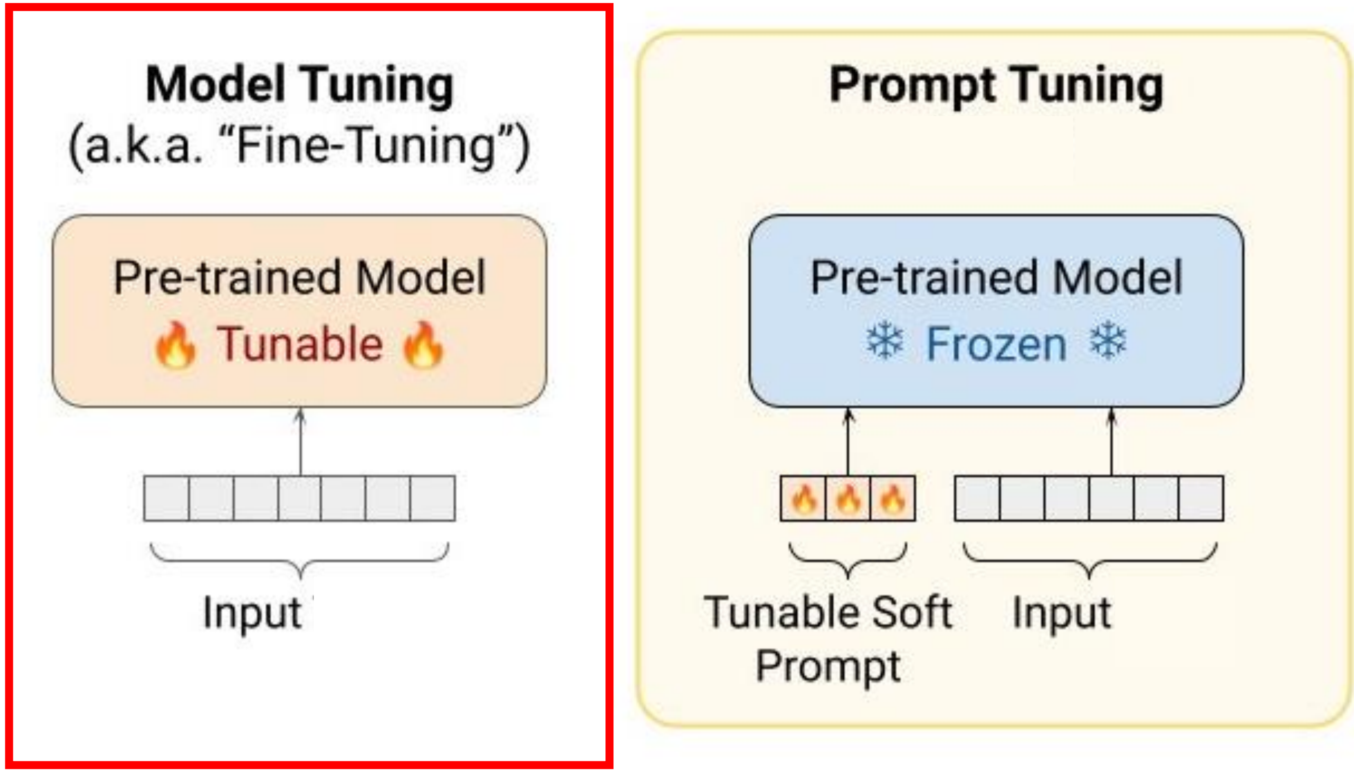
B ďk

- ďk ďkďk ďk ďkďk
- ďkďk ďk ďk ďk ďk ďk



B ck

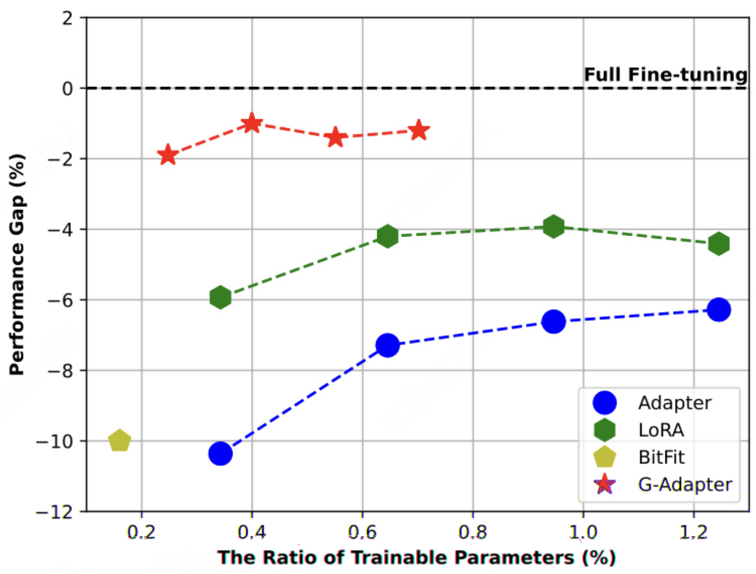
- ck ckck ck ckck
    - ckck ck ck ck ck ckck



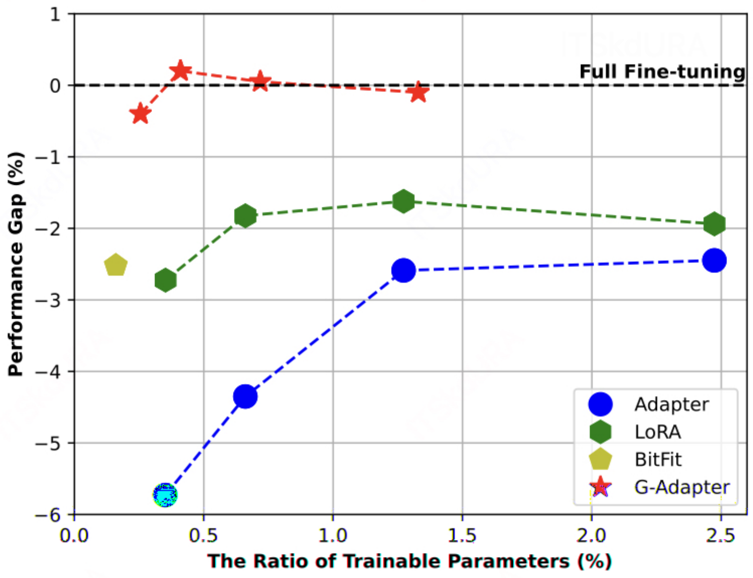
Can PEFTs from the language domain be transferred directly to graph-based tasks?

- There is a significant gap between traditional PEFTs and full fine-tuning, especially on large-scale datasets.

How to design a graph-specific PEFT method?



(a) On large-scale datasets.

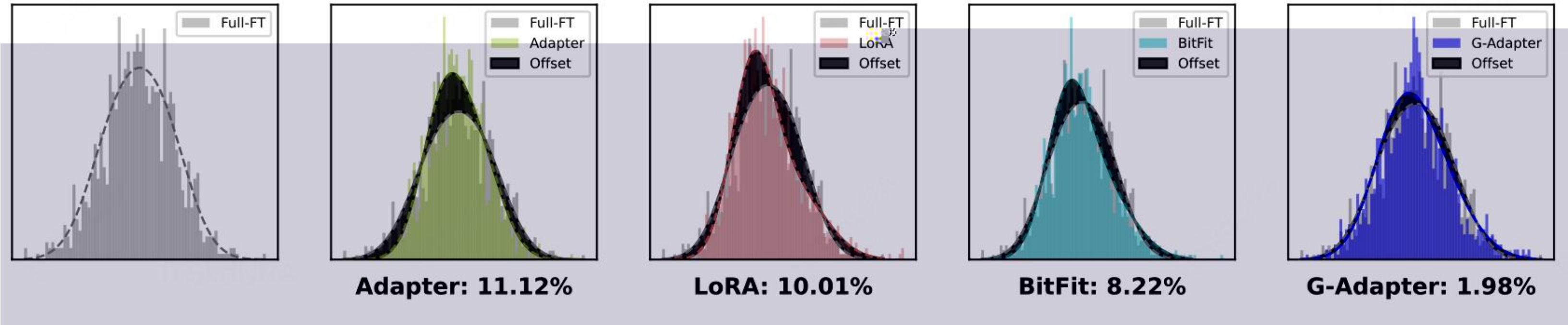


(b) On small-scale datasets.

Gui, A., Ye, J., & Xiao, H. G-adapter: Towards structure-aware parameter-efficient transfer learning for graph transformer networks. AAAI2024

ck

- ck ck ck ck ck ck ck ck ck ck ck
  ck ck ck ck
- ck ck ck ck ck ck ck ck ck ck
  ck ck ck ck ck ck ck
  ck ck ck ck ck
- ck ck ck ck ck ck ck ck
  ck ck



- Delta tuning improves the traditional fine-tuning in the catastrophic forgetting of pre-trained knowledge problem and overfitting problem.

How to effectively utilize the advantages of delta tuning while preserving the expressivity of GNNs?

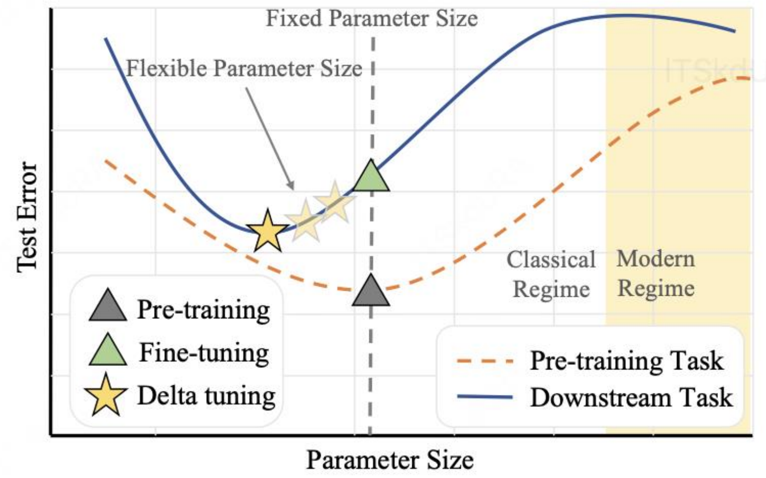
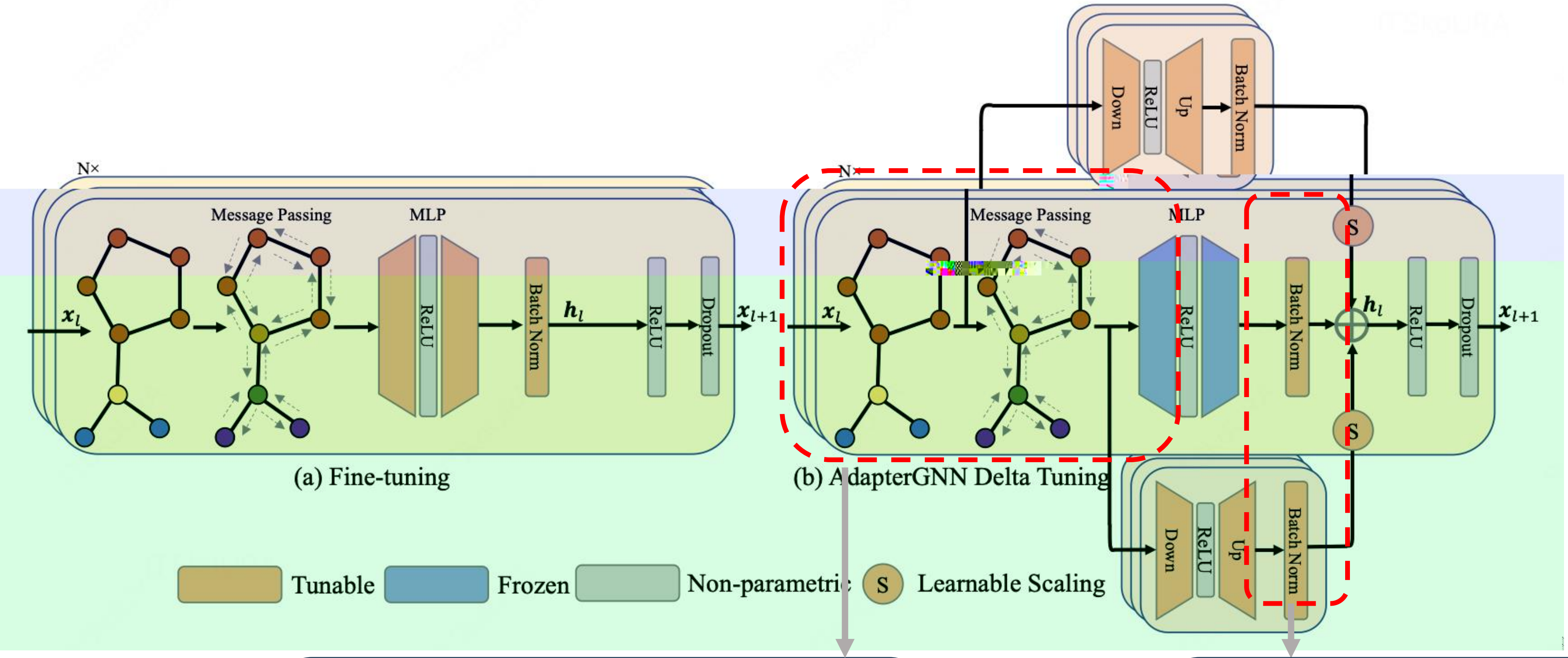


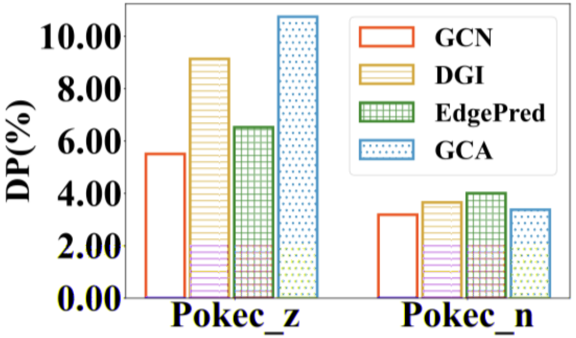
Figure 1: A large model is often employed for pre-training ▲ when sufficient data is available. However, for downstream tasks with limited data, a smaller model is optimal in the classical regime. Compared with fine-tuning ▲, delta tuning ★ preserves expressivity while reducing the size of parameter space, leading to lower test error.

Li, S., Han, X., & Bai, J. AdapterGNN: Parameter-Efficient Fine-Tuning Improves Generalization in GNNs. AAAI2024

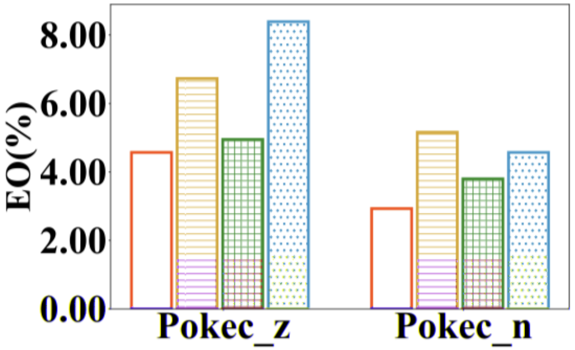

N×
Message Passing
MLP
ReLU
Batch Norm
h_l
ReLU
Dropout
x_l
x_{l+1}
(a) Fine-tuning
(b) AdapterGNN Delta Tuning
Down
ReLU
Up
Batch Norm
S
Tunable
Frozen
Non-parametric
Learnable Scaling

- Recent works have demonstrated that pre-trained language models tend to inherit bias from pre-training corpora.
- Pre-trained Graph Models(PGMs) can well capture semantic information on graphs during the pre-training phase, which inevitably contains sensitive attribute semantics.

How to improve the fairness of PGMs?



(a) Demographic Parity (DP).



(b) Equality Opportunity (EO).

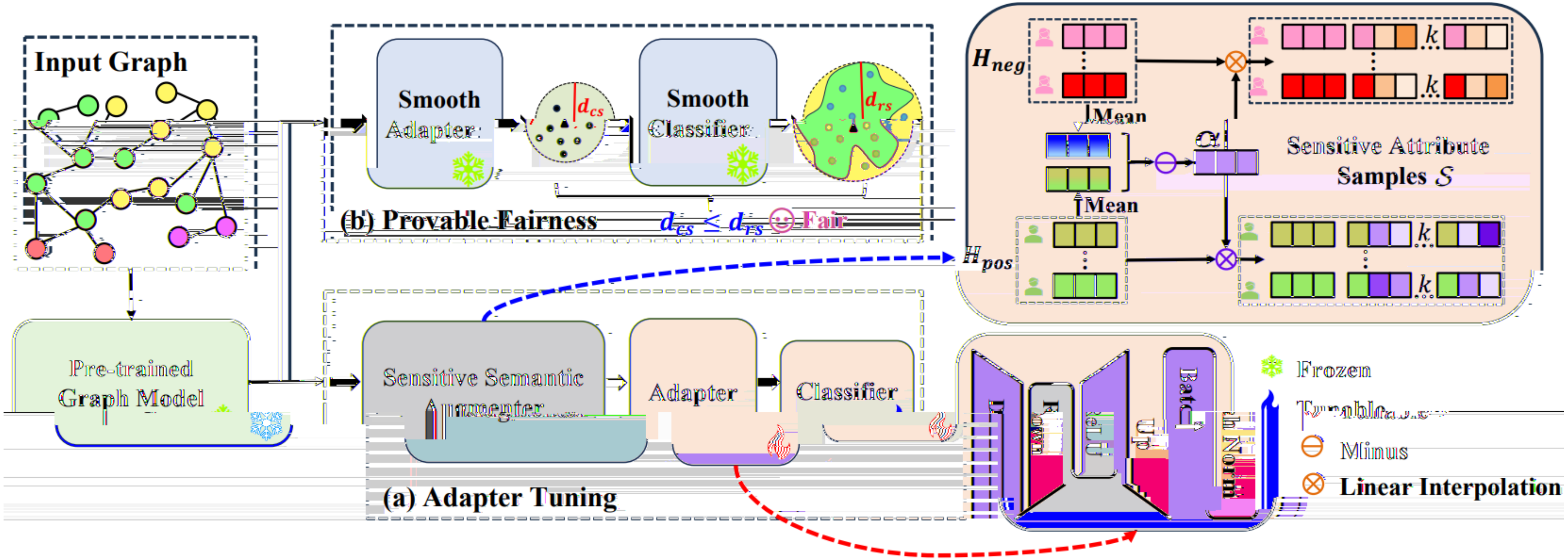
Zhang, Z., Zhang, M., Yu, Y., Yang, C., Liu, J., & Shi, C. Endowing Pre-trained Graph Models with Provable Fairness. WWW2024 ck ck ck ck ck

- Existing works generally train a fair GNN for a specific task.
- Debiasing for a specific task in the pre-training phase is inflexible, and maintaining a specific PGM for each task is inefficient.

ck ck ck ckck

- Provable lower bounds on the fairness of model prediction.

How to efficiently and flexibly endow PGMs fairness with practical guarantee?



$$\boldsymbol{\alpha} = \mathbf{h}_{pos} - \mathbf{h}_{neg},$$

$$\mathbf{h}_{pos} = \frac{1}{n_{pos}} \sum_{i=1}^{n_{pos}} \mathbf{H}_{pos,i} \, , \mathbf{h}_{neg} = \frac{1}{n_{neg}} \sum_{i=1}^{n_{neg}} \mathbf{H}_{neg,i}$$

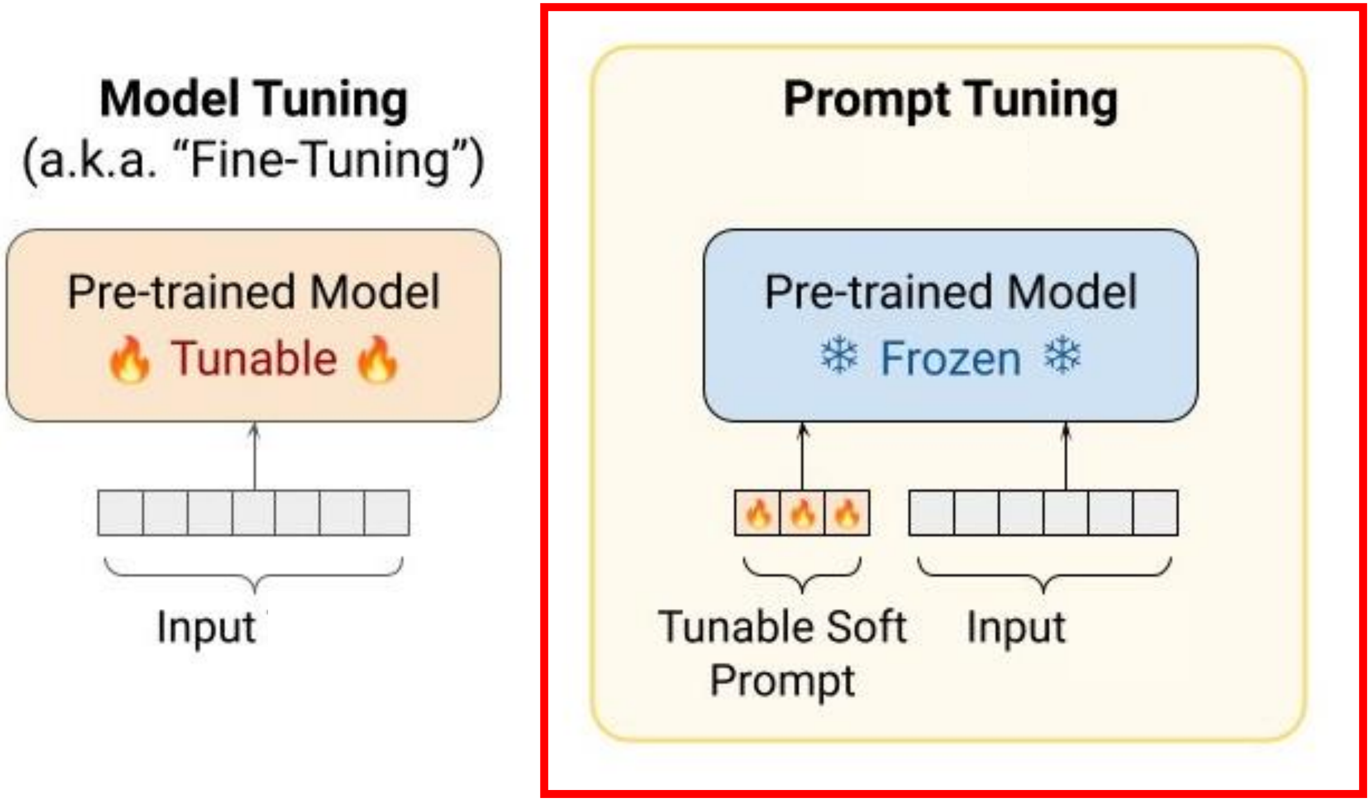
$$\mathcal{S}_i := \{\mathbf{h}_i + t \cdot \boldsymbol{\alpha} \mid |t| \leq \epsilon\} \subseteq \mathbb{R}^p,$$

$$\mathcal{L}_{\text{RandAT}} = \mathbb{E}_{i \in \mathcal{V}_L} \left[\mathbb{E}_{\mathbf{h}_i' \in \hat{\mathcal{S}}_i} \left[\ell(d \circ g(\mathbf{h}_i'), y_i) \right] \right],$$

$$\mathcal{L}_{MinMax}(\mathbf{h}_i) \approx \max_{\mathbf{h}_i' \in \hat{\mathcal{S}}_i} \left\| g(\mathbf{h}_i) - g(\mathbf{h}_i') \right\|_2 .$$

B ck

- ckck ck ck ck
  - ck
  - ck ck

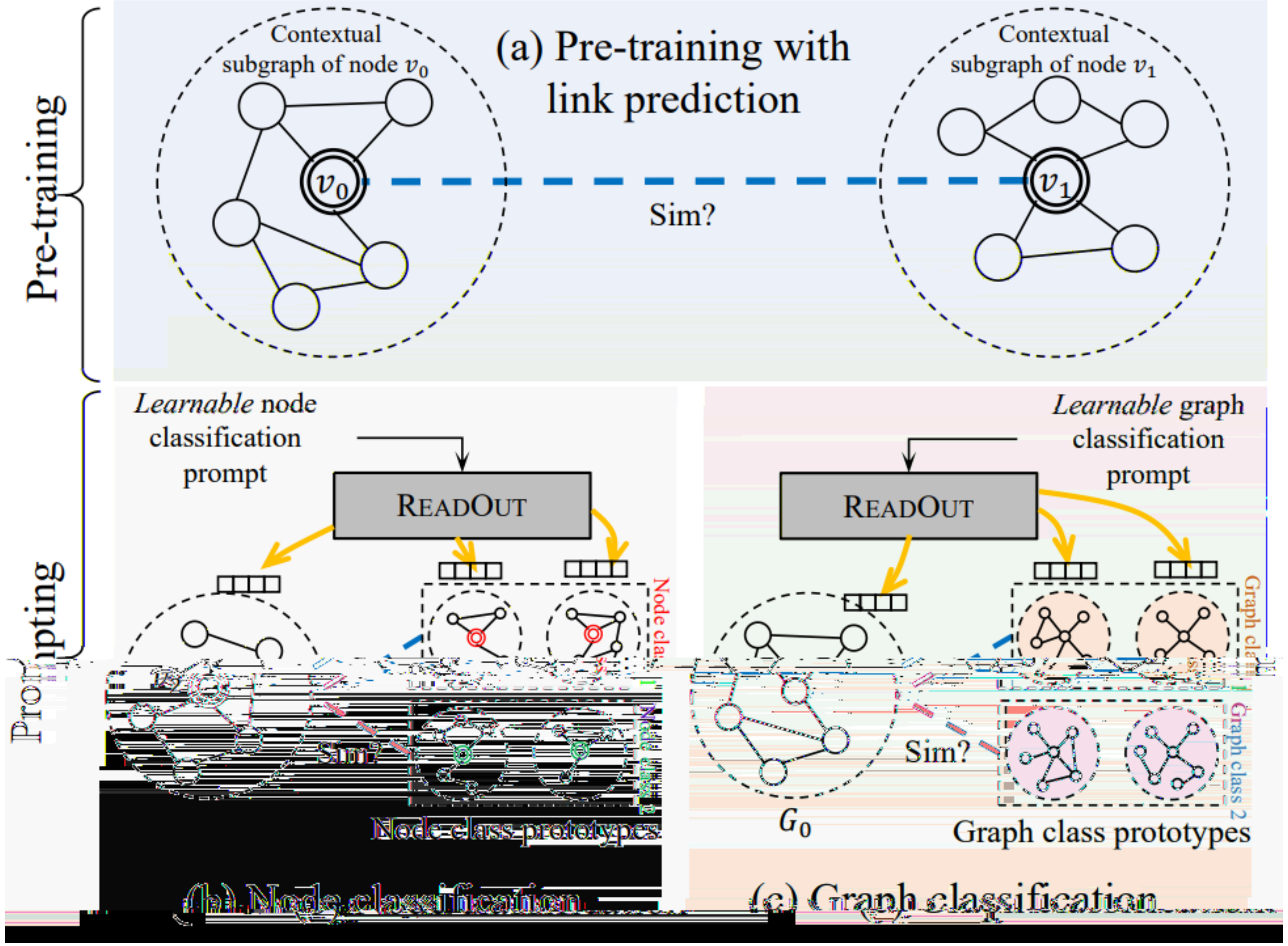


ck

- How to unify various pre-training and downstream tasks on graph?
- How to design prompts on graph?

:

- A unified task template based on subgraph similarity computation
- Use a learnable prompt to guide graph readout for different tasks



Liu, Z., Yu, X., Fang, Y., & Zhang, X. (2023, April). Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. WWW2023

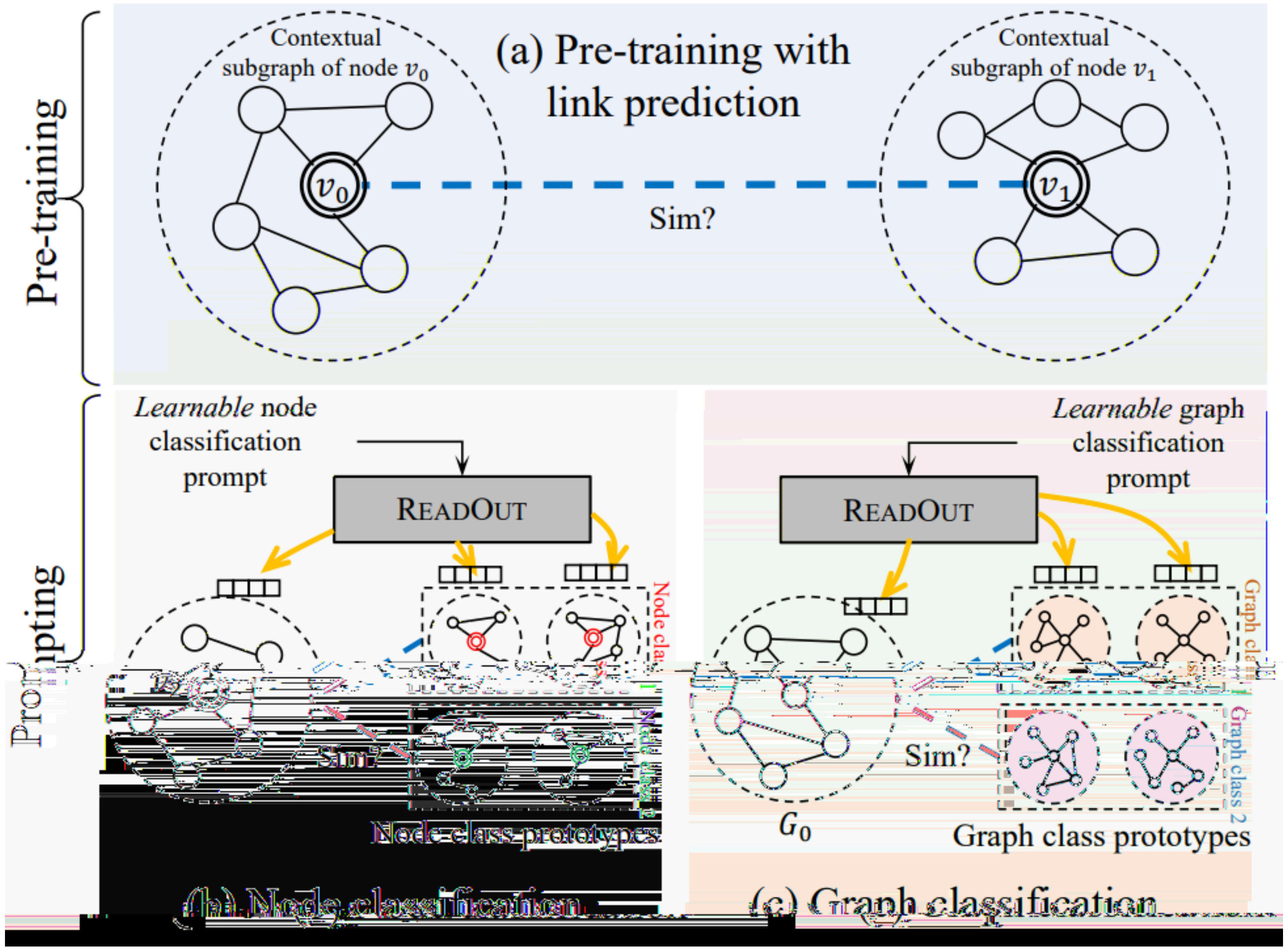
ck

Different downstream tasks require different subgraph readout
→ Use task-specific learnable prompts

Prompt vector added to the readout layer of the pre-trained GNN

$$\mathbf{s}_{t,x} = \text{ReadOut}(\{\mathbf{p}_t \odot \mathbf{h}_v : v \in V(S_x)\})$$

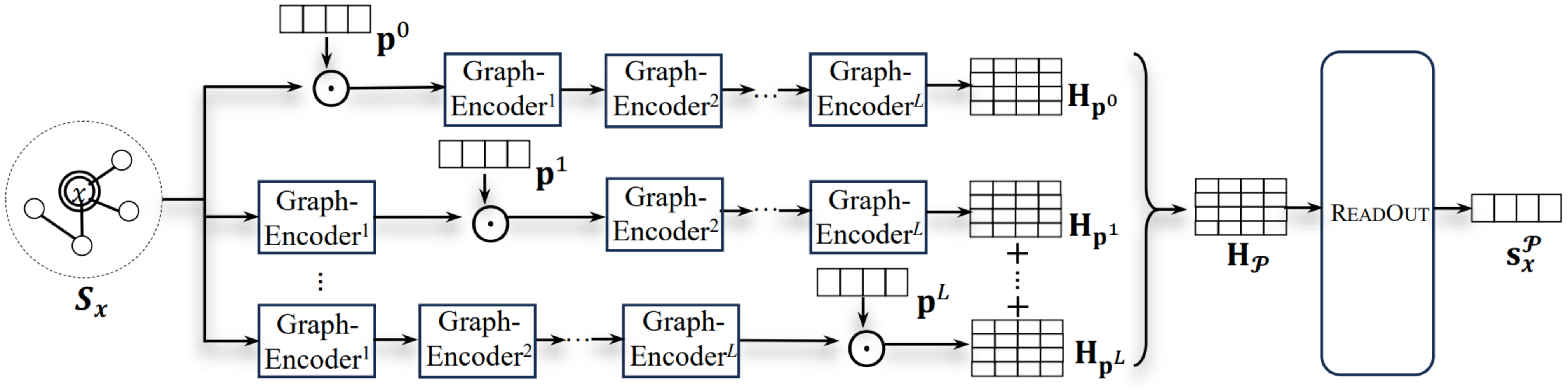
$\mathbf{s}_{t,x}$: (sub)graph embedding of x for a task t
$\mathbf{h}_v$: node v's embedding vector
$\mathbf{p}_t$ or $\mathbf{P}_t$: learnable prompt vector or matrix for task t



P ck ck ck ck :

- B

ck ck



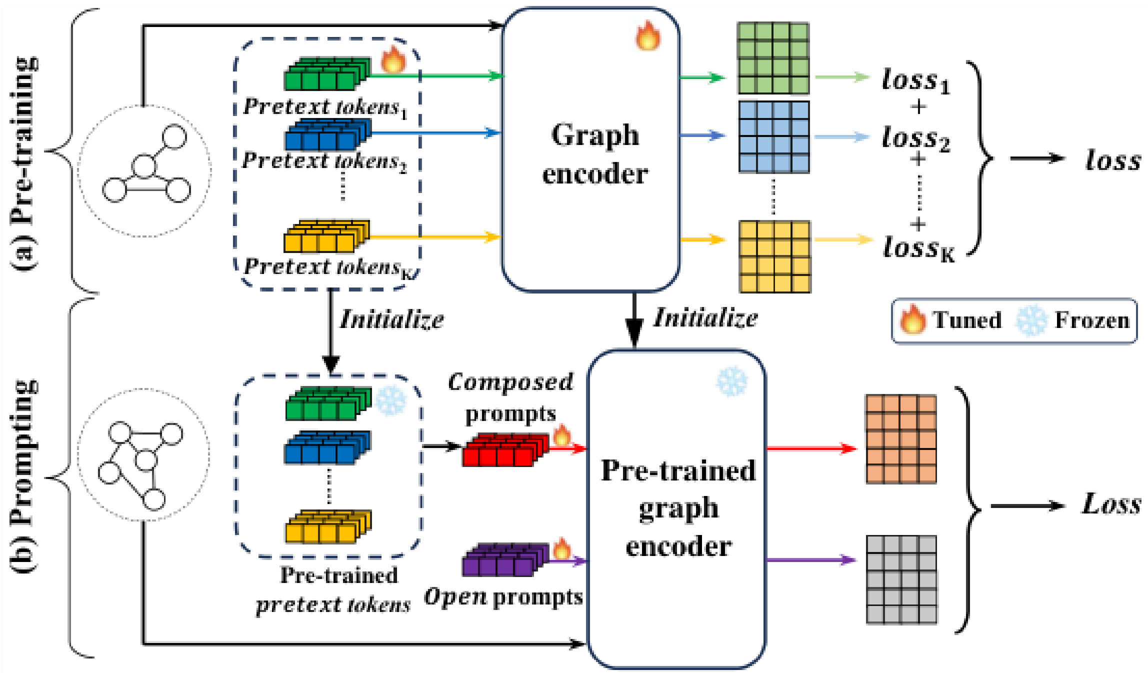
Yu, X., Liu, Z., Fang, Y., Liu, Z., Chen, S., & Zhang, X. Generalized graph prompt: Toward a unification of pre-training and downstream tasks on graphs. arXiv preprint arXiv:2311.15317.

ƈk

- To cater to diverse downstream tasks, pre-training should broadly extract knowledge from various aspects.

ƈk ƈk：

- Different pretext tasks often have different objectives, directly combining them lead to task interference.
- Multiple pretext tasks further complicates the alignment of downstream objectives with the pre-trained model.



C1: How can we leverage diverse pre-text tasks for graph models in a synergistic manner?

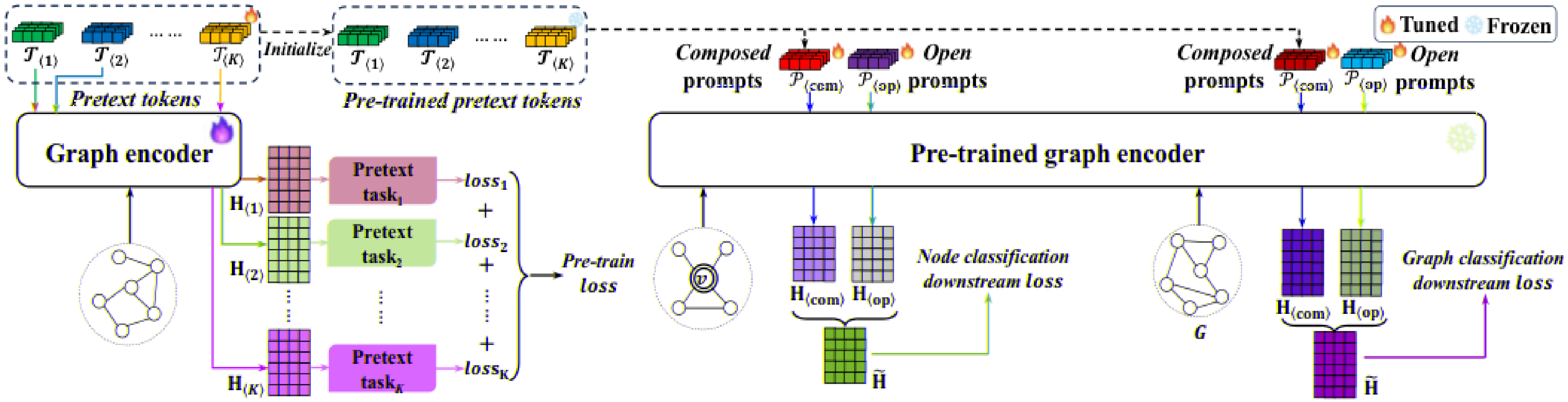
C2: How can we transfer both task-specific and global pre-trained knowledge

Multi-task pre-training

$$\mathcal{T}_{\langle k\rangle} = \{\mathbf{t}_{\langle k\rangle,0}, \mathbf{t}_{\langle k\rangle,1}, \ldots, \mathbf{t}_{\langle k\rangle,L}\}$$

$$\mathbf{H}^{l+1} = \mathrm{MP}(\mathbf{t}_{\langle k\rangle,l} \odot \mathbf{H}^{l}, \mathbf{A}; \theta^{l})$$

$$\mathbf{H}_{\mathbf{t}} = \textsc{GraphEncoder}_{\mathbf{t}}(\mathbf{X}, \mathbf{A}; \Theta)$$



(a) Multi-task pre-training
(b) Downstream node classification
(c) Downstream graph classification

Prompt tuning

$$\mathcal{P}_{\langle \mathrm{com}\rangle} = \{\mathbf{p}_{\langle \mathrm{com}\rangle,0}, \mathbf{p}_{\langle \mathrm{com}\rangle,1}, \ldots, \mathbf{p}_{\langle \mathrm{com}\rangle,L}\}$$

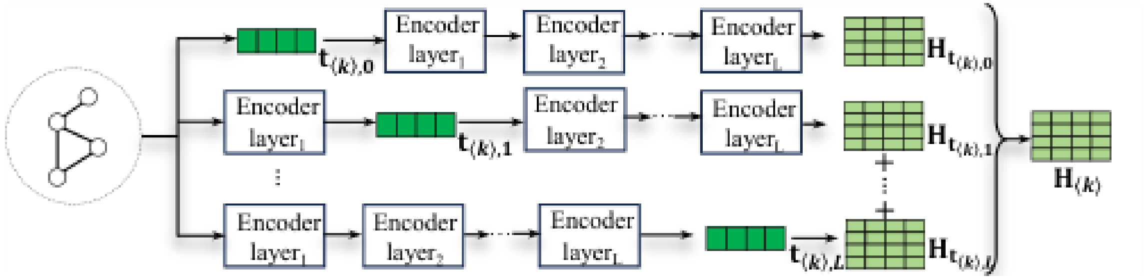
$$\mathbf{p}_{\langle \mathrm{com}\rangle,l} = \textsc{Compose}(\mathbf{t}_{\langle 1\rangle,l}, \mathbf{t}_{\langle 2\rangle,l}, \ldots, \mathbf{t}_{\langle K\rangle,l}; \Gamma)$$

$$\mathcal{P}_{\langle \mathrm{op}\rangle} = \{\mathbf{p}_{\langle \mathrm{op}\rangle,0}, \mathbf{p}_{\langle \mathrm{op}\rangle,1}, \ldots, \mathbf{p}_{\langle \mathrm{op}\rangle,L}\}$$

$$\mathbf{H}_{\mathbf{p}} = \textsc{GraphEncoder}_{\mathbf{p}}(\mathbf{X}, \mathbf{A}; \Theta_{\mathrm{pre}})$$

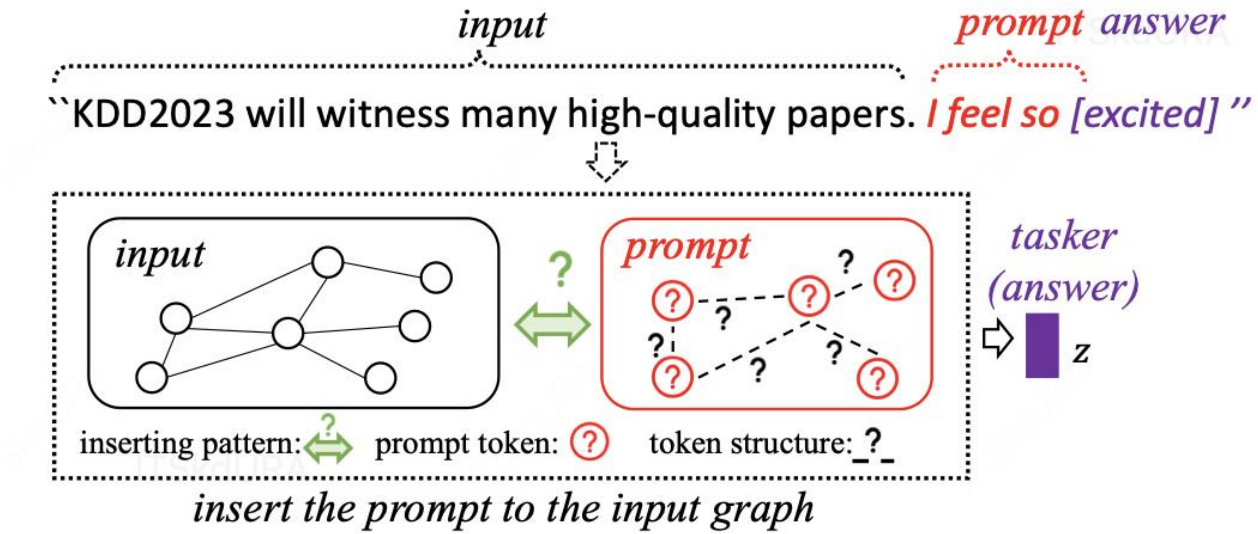
$$\tilde{\mathbf{H}} = \textsc{Aggr}(\mathbf{H}_{\langle \mathrm{com}\rangle}, \mathbf{H}_{\langle \mathrm{op}\rangle}; \Delta)$$

$$\mathbf{H}_{\langle k\rangle} = \sum_{l=0}^{L} \alpha_l \mathbf{H}_{\mathbf{t}_{\langle k\rangle,l}} \qquad \mathcal{L}_{\mathrm{pre}}(\mathcal{H}; \mathcal{T}, \Theta) = \sum_{k=1}^{K} \beta_k \mathcal{L}_{\mathrm{pre}_{\langle k\rangle}}(\mathbf{H}_{\langle k\rangle}; \mathcal{T}_{\langle k\rangle}, \Theta),$$



ƈk ƈk：

- Graph prompt not only requires the prompt “content” but also needs to know how to organize these tokens and how to insert the prompt into the original graph.
- There is a huge difficulty in reconciling downstream problems to the pre-training task.
- Learning a reliable prompt needs huge manpower and is more sensitive in multi-task setting.

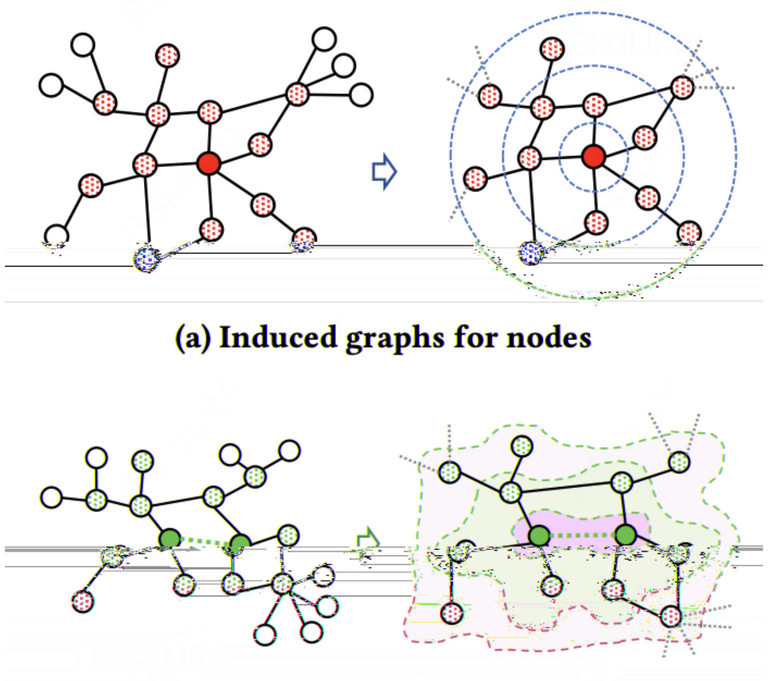


ƈk ƈkB ƈk :

- This work reformulates node-level and edge-level tasks to graph-level tasks by building induced graphs for nodes and edges, respectively.

Bƈk :

- This work introduces some prompt nodes with unique connection relationships between them and adaptively insert them into the original input graph, in order to obtain a prompt graph.


(a) Induced graphs for nodes

(b) Induced graphs for edges

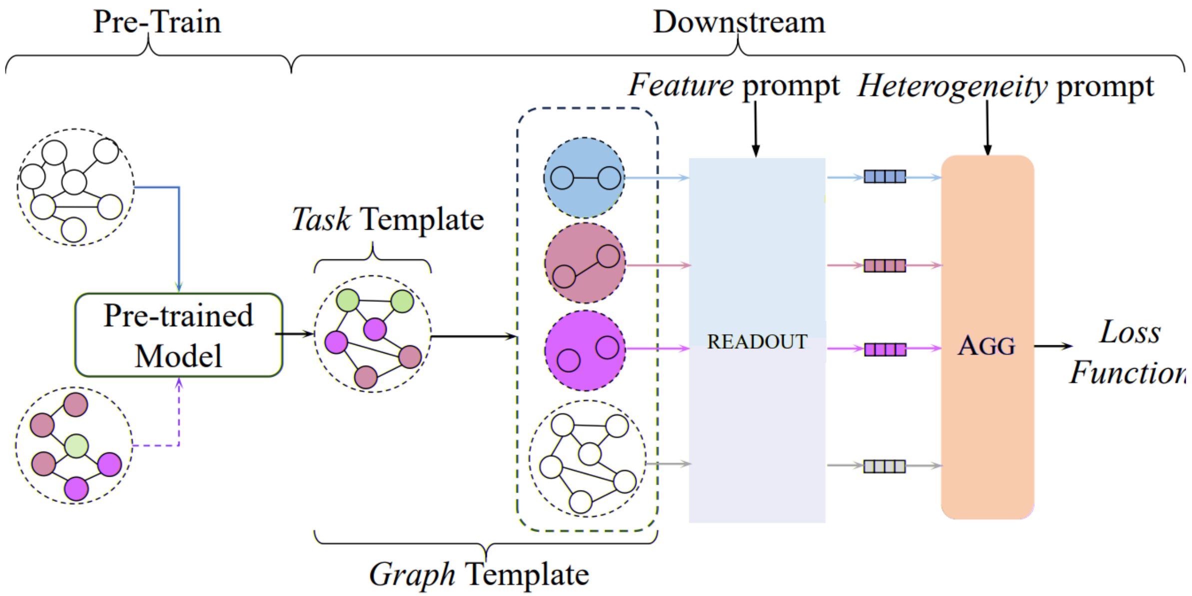
ɗk

- Gap between homogeneous and heterogeneous graph.
- Different downstream tasks focus on heterogeneous aspect.

:

- B ɗk ɗk ɗk ɗk

- B ɗk ɗk ɗkɗk ɗk ɗk



(a) Target scenarios (b) *Dual*-template (c) *Dual*-prompt

Yu, X., Fang, Y., Liu, Z., & Zhang, X. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. AAAI2024

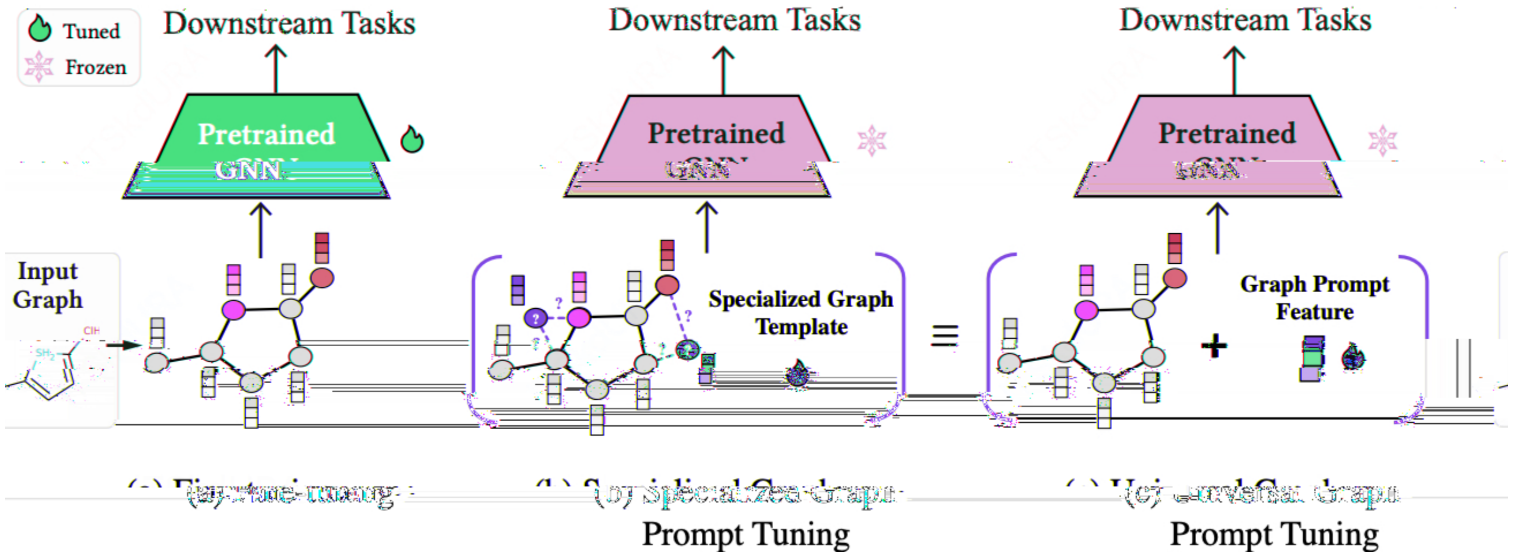
ck ck：

- Diverse pre-training strategies employed on graphs make it difficult to design suitable prompting functions.
- Existing prompt-based tuning methods for GNN models are predominantly designed based on intuition, lacking theoretical guarantees for their effectiveness.

Fang, T., Zhang, Y., Yang, Y., Wang, C., & Chen, L. Universal prompt tuning for graph neural networks. Neurips2023

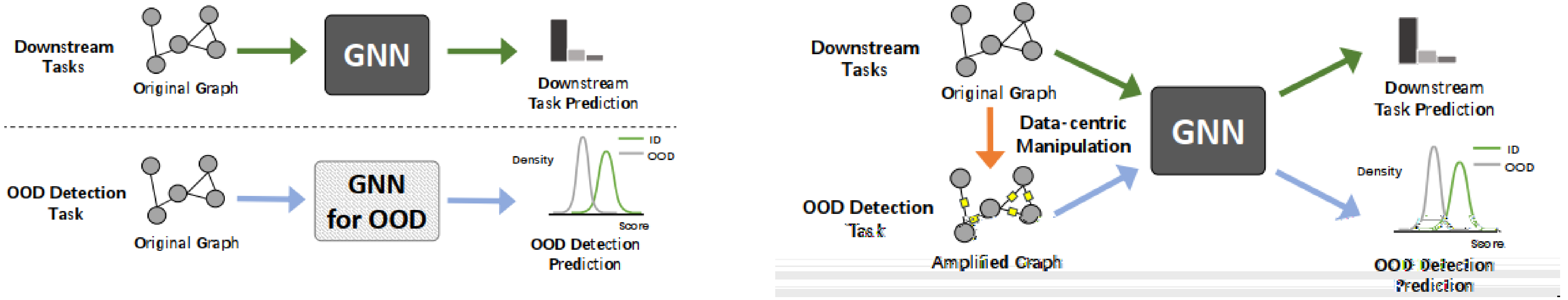
ck :

- This work proposes a universal prompt-based tuning method that can be applied to the pre-trained GNN models that employ any pre-training strategy.
- GPF operates on the input graph's feature space and involves adding a shared learnable vector to all node features in the graph.
- GPF-plus is a theoretically stronger variant of GPF, for practical application, which incorporates different prompted features for different nodes in the graph.



- A reliable GNN should not only perform well on know samples (ID) but also identify graphs it has not been exposed to before (OOD) .
- Existing works proposes to train a neural network specialized for the OOD detection task.

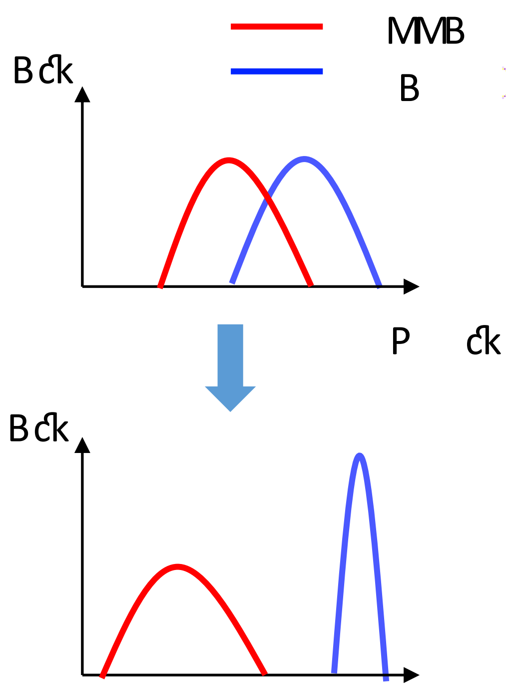
Can we build a graph prompt that can solve OOD detection given a well-trained GNN?

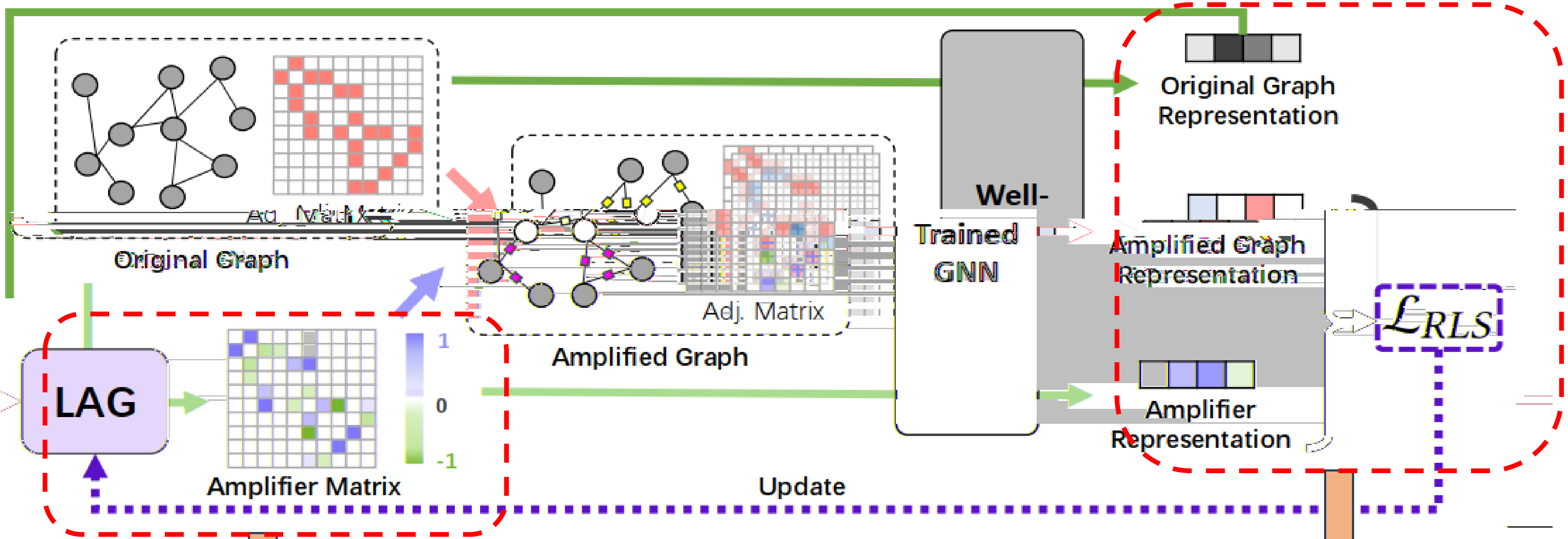


(1) Traditional works

(2) Our proposed framework

V V ck V P B B ck ck L ck L ck M M B Bckck BB

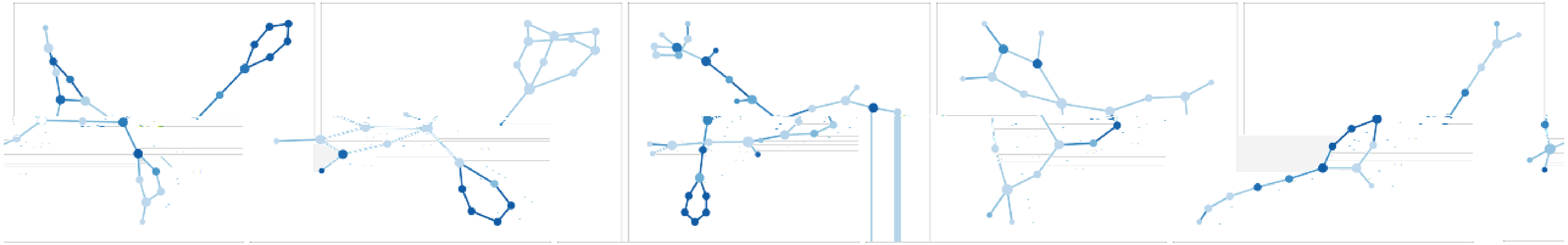

MMB
B
P
P


Original Graph
Amplified Graph
Adj. Matrix
LAG
Amplifier Matrix
1
0
-1
Well-Trained GNN
Original Graph Representation
Amplified Graph Representation
Amplifier Representation
$\mathcal{L}_{RLS}$
Update

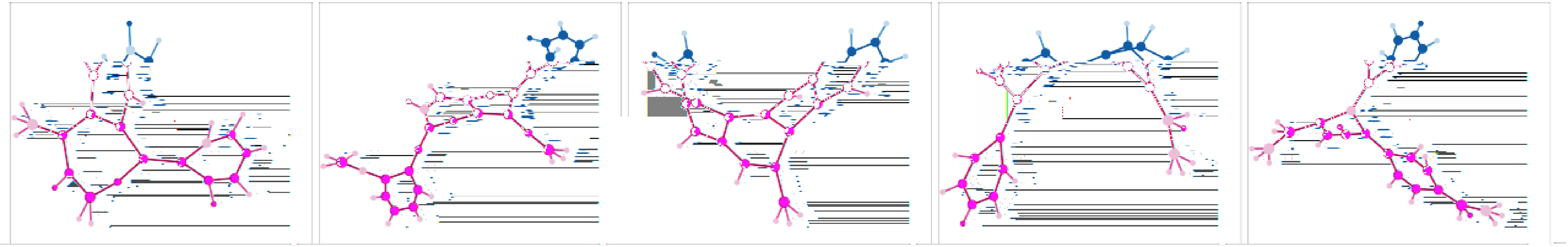
We conducted experiments on five dataset pairs over four GNNs to verify performance.

ID	OOD	Metric	GCL_S	GCL_S+	Improv.	GCL_L	GCL_L+	Improv.	$JOAO_S$	$JOAO_S$+	Improv.	$JOAO_L$	$JOAO_L$+	Improv.
ENZYMES	PROTEIN	AUC ↑	62.97	**73.76**	+17.14%	62.56	**67.15**	+7.34%	61.20	**74.19**	+21.23%	59.68	**65.11**	+9.10%
		AUPR ↑	62.47	**75.27**	+20.49%	**65.45**	65.18	-0.41%	61.30	**77.10**	+25.77%	64.16	**64.49**	+0.51%
		FPR95 ↓	93.33	**88.33**	-5.36%	93.30	**85.00**	-8.90%	90.00	**81.67**	-9.26%	96.67	**85.00**	-12.07%
IMDBM	IMDBB	AUC ↑	80.52	**83.84**	+4.12%	61.08	**68.64**	+12.38%	80.40	**82.80**	+2.99%	48.25	**64.32**	+33.31%
		AUPR ↑	74.43	**80.16**	+7.70%	59.52	**68.03**	+14.30%	74.70	**77.77**	+4.11%	47.88	**61.62**	+28.70%
		FPR95 ↓	38.67	**38.33**	-0.88%	96.67	**91.33**	-5.52%	44.70	**42.00**	-6.04%	98.00	**94.00**	-4.08%
BZR	COX2	AUC ↑	75.00	**97.31**	+29.75%	34.69	**65.00**	+87.37%	80.00	**95.25**	+19.06%	41.80	**65.62**	+56.99%
		AUPR ↑	62.41	**97.17**	+55.70%	39.07	**62.89**	+60.97%	67.10	**94.34**	+40.60%	56.70	**67.22**	+18.55%
		FPR95 ↓	47.50	**15.00**										

Case study: We visualize the learned graph prompts (i.e., amplifiers) for interpretability analysis.

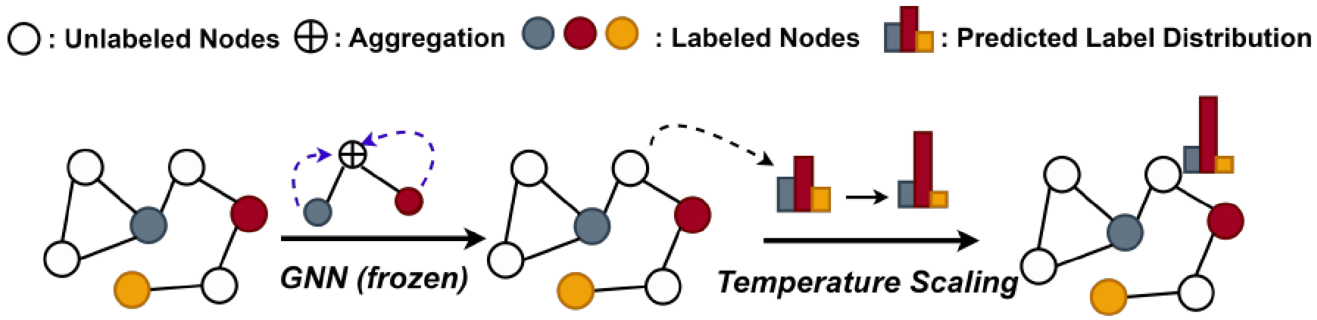


(a) ID (b) ID (c) ID (d) OOD (e) OOD



(a) ID (b) ID (c) ID (d) OOD (e) OOD

- Existing calibration methods focus on improving GNN models. Recent work has shown that the post-hoc methods, such as temperature scalling-based calibration, can achieve a better trade-off between accuracy and calibration.



(a) Temperature scaling-based calibration

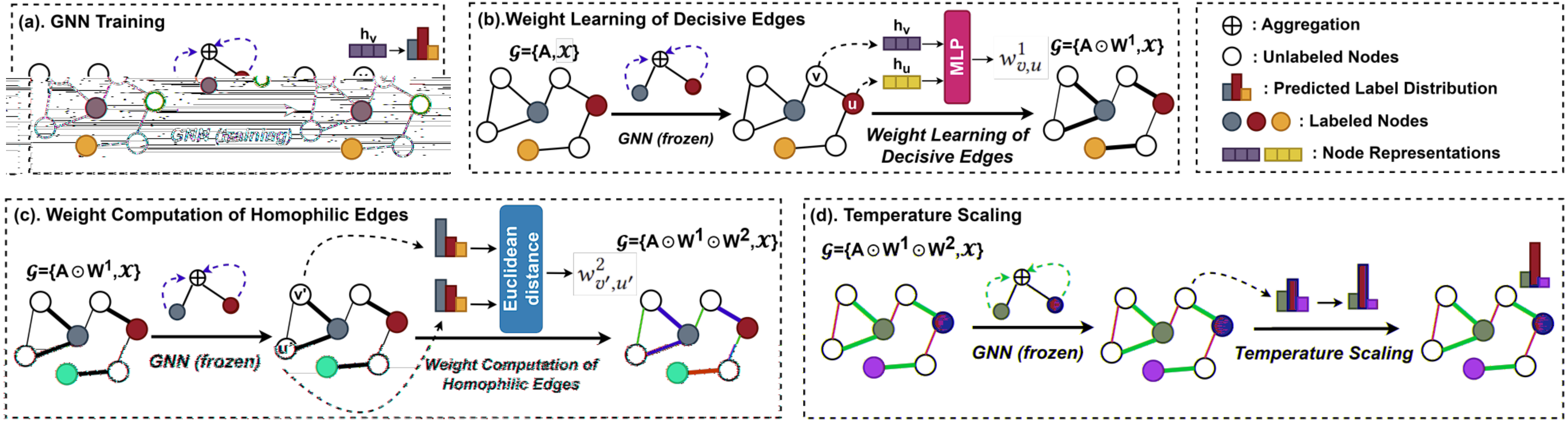
- Through evaluating the expected calibration error (ECE) on Cora and Photo datasets with five different GNNs, we find that the ECEs on Cora (10.25%-18.02%) are always larger than those on Photo (4.38%-8.27%), indicating that **the calibration performance depends more on the datasets instead of GNN model**.

- ɗk ɗk ɗk ɗk ɗk ɗk ɗk L L
  ɗk ɗk ɗk ɗk

Can we modify the graph data instead for better calibration performance without losing accuracy?

(b) Data-centric calibration

- We propose Data-centric Graph Calibration (DCGC) with two edge weighting modules to adjust the input graph.



- ck
  - ckck ck ck
  - ck ck ck ck ck ck ck ck
  - ck ck ck ck ck ck
- B ck
  - ck ck ck ck ck ck ck ck

    ck ck ck ck ck
  - ck ck ck ck ck